Government MCA College

Maninagar (East), Ahmedabad

VISION:

Provide value-based quality education for computer science applications which enable students to solve real-life problems of society.

MISSION:

- To equip our students with good knowledge, skills and attitude to solve real-life problems in the domain of computer applications.
- To establish industry-academia interaction to facilitate the students to work proficiently in the industrial environment.
- To imbibe high moral values and professional ethics.
- To provide a conducive environment so as to achieve excellence in teaching-learning, and research and development activities.

Government MCA College

Maninagar (East), Ahmedabad

Program Objectives (POs):

- PO1 (Foundation Knowledge) : Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.
- PO2 (Problem Analysis) : Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.
- PO3 (Development of Solutions) : Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.
- PO4 (Modern Tool Usage) : Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
- PO5 (Individual and Teamwork) : Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use

methodologies such as agile.

- PO6 (Project Management and Finance) : Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
- PO7 (Ethics): Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware
- PO8 (Life-long learning) : Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

Government MCA College

Maninagar (East), Ahmedabad

Program Educational Objectives (PEOs):

- Identify and analyze real-life problems and design computing systems appropriate to their solutions that are technically sound, economically feasible, and socially acceptable.
- Exhibit professionalism, ethical attitude, good communication skills, and teamwork in their profession.
- Adapt to current trends by engaging in life-long learning.

Program Specific Outcomes (PSOs):

- Meet the needs of users within an organizational and societal context through the selection, creation, application, integration, and administration of computing technologies.
- Apply concepts, probability, statistics, and mathematics through calculus (differential and integral), numerical methods, and sciences, including applications appropriate to the field of computing problems.
- Use algorithms, data structures, database management, software design, concepts of programming languages, and computer organization and architecture in computer applications.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Problem Solving using C
Subject Code: 619401

With effective from academic year 2020-21

1. Learning Objectives:

The student will be able to develop solution of problem using his/her own logic and gradually learn to solve various problems using efficient algorithm and Programming techniques.

2. **Prerequisites:** Logical Thinking Ability, School-level mathematics

3. **Mode of Delivery**:
In tutorial sessions, discuss the basics and the methodology for tasks to be completed during next lab session(s). In the Lab sessions – Ask students to write, compile and test the programs on computer. Check for common difficulties faced by students. Conduct a brief session to explain and then resume the hands-on working. If a few students are not able to complete all the assigned task(s) during lab sessions, help them and motivate them to complete pending tasks at home.

4. Course Contents:

Tasks #	***Topics (Programs) to be Completed***	***Est. Hrs.***
1.	* Introduction: Objectives; Methodology; C Language Constructs; Good Programming Practices Write, compile and execute the following programs: * Simple programs-1: Print “Hello world!”; Print “My name is <Name>”; Print “<Address Lines>” Explain the concept of variables, their declaration, and Arithmetic Assignment statements * Simple programs-2a: x1 = 2; x2 = 7; Print “x1 = value, y1 = value”. * Write a program to take 5 integers and find and print the total and average of the 5 numbers. Repeat the same for floating point numbers instead of integers. Explain for-loop. Also explain the general algorithm for summation: (i) Initialize sum = 0; (ii) repeat sum = sum + value * Write a program to find the sum of 1, 2, 3, … , n. Print average (avg) also. Explain scanf() for accepting user inputs. * Write a program to accept n. Find sum of n values accepted 1-by-1. Also find average (avg). Print sum, avg. Additionally, print the input values also. Explain Arrays to store multiple values * Write a program to accept n and n input values to be stored in an array. Find sum and average (avg) of n values. Print input values followed by sum, avg.	2 + 4
2.	Explain string variable, its declaration and scanf() to accept strings	2 + 4

	* Write a program to accept as input: first name, middle name, surname; then print name, first as (a) first mid surname; and thereafter (b) surname first mid, etc in Explain algorithm to find string length. Explain: if-statements: if ( ) {… }; if ( ) {… } else {… }; and if ( ) {… } else if ( ) {… }; * Write a program to find string length. Is the string length same as number of characters in the string? Explain also the concept of function. * Rewrite the program to find string length by using function for finding string length. Test this program to find lengths of first, mid, and surname. * Write a program to print a given string in reverse order.	
3.	Explain: Convert temperature in degrees Fahrenheit to degrees Celsius and vice versa. * Write a program to take Input: n, and n values of temperature in ^{o}F, convert these into ^{o}C and print the values in a table with 1st column containing ^{o}F and 2nd column ^{o}C Explain the concept of function() in C. * Modify the above program to convert temperature in ^{o}C into ^{o}F. Write a function c2f() for this operation. How many input arguments are there? How many outputs from the function?	1 + 2
4.	Explain the method of checking of "Divisibility of an integer by another integer" – Modulus operator (%). For segregating a given list of integers into even and odd numbers using bitwise operations. * Write a program in C: Inputs: 2 integer values: Numerator (num1) and Denominator (num2). Output: Quotient (q) and Remainder (r). * Write a program in C: Input: An array (a List) of n integers. Output: * Develop an algorithm write a program in C to print all ***primes*** in the first n (n > 1) integers. Develop the most efficient algorithm.	2 + 4
5.	Explain Euler's algorithm to find gcd(m, n). * Given two integers m and n (m, n >= 0), develop an algorithm and write a program in C to find their greatest common divisor (gcd) * Write a program in C to rearrange the elements in an array so that they appear in reverse order.	2 + 4 Cum. 9+18
6.	* Write a program to calculate and display the value of the slope of the line connecting the two points whose coordinates are (3, 7) and (8, 12). Slope of a line between two points (x_1 , y_1) and (x_2 , y_2) is ($y_2 – y_1$) / ($x_2 – x_l$). Run the same program for the line connecting the points (2, 10) and (12, 6), and other pairs of points.	1 + 2

	* Write a program to calculate and display the coordinates of the midpoint of the line connecting the two points given in the previous Exercise. The coordinates of the midpoint between two points having coordinates (x_1 , y_1) and (x_2 , y_2) are ((x_1 + x_2) / 2, (y_1 + y_2) / 2). * Write a program that calculates the distance between two points whose coordinates are (7, 12) and (3, 9). Distance between two points having coordinates (x_1 , y_1) and (x_2 , y_2) = sqrt($[x_1 - x_2]^2 + [y_1 - y_2]^2$). Also, run the program for the points (-12, -15) and (22, 5) and a few other points.	
7.	Introduce O(n) notation as an indicator of how fast a program works. * Given some integer x, develop an algorithm and write a program to compute the value of x^n where n is considerably larger than 1. This algorithm has time complexity O(n). * Develop an improved algorithm having time complexity O($\log_2 n$).	2 + 4
8.	Explain: (a) sin(x) defined by the infinite series expansion. What will be the stopping criterion? * Write a program to evaluate sin(x) * Write a program to determine and display the maximum height reached when the ball is thrown at 5 miles / hour at an angle of 60 degrees. (Hint: Make sure to convert the initial velocity into the correct units.) The maximum height reached by a ball thrown with an initial velocity v in feet/sec at an angle of θ is given by the formula height = ($0.5 * v^2 * \sin^2 \theta$) / 32.2. Run the program for v = 7 miles / hour and angle = 45 degrees. Explain the problems and the solution approach. * Write a program to calculate and print the height (h) = L * sin θ , where L is the Length of the Ladder, and θ is the angle the ladder makes with the horizontal base. Data: (a) L = 20 feet, θ = 85° , (b) L = 25 feet, θ = 75° . * Write a program that calculates the x and y coordinates of the point whose polar coordinates are r = 10 and θ = 30° , using the following formulas: x = r cos θ and y = r sin θ . Run the program again to convert polar coordinates: r = 12.5 and θ = 67.8° into rectangular coordinates.	2 + 4
9.	* Given an integer n >= 1, develop an algorithm and write a program to find the smallest exact divisor of n other than one. * Every integer can be expressed as a product prime numbers. Develop an algorithm and write a program to compute all the prime factors of a given integer n > 0.	1 + 2 Cum. 15+30

10.	* Write a program in C (a) to find factorial of n (n!), and (b) first n terms of the Fibonacci sequence using an iterative algorithm. Explain Recursion formulation of a problem and its working by taking 3 examples: (a) n! ; (b) fibo(n); and (c) sum of $a_0 + a_1 + \ldots + a_{n-1} + a_n$. * Write a program in C (a) to find factorial of n (n!); (b) first n terms of the Fibonacci sequence using an iterative algorithm; and (c) sum of $a_0 + a_1 + \ldots + a_{n-1} + a_n$.	2 + 4
11.	Explain: Exchange the values of two variables. * Program : Input: a, b Output: a (prints the value of b), b (prints the value of a) Explain preliminary concept of pointers and write a few simple programs to help understanding the concept of ***pointers***: * Implement the program in C for "exchanging the values of two variables" using function (which will require use of pointers for function arguments in C) * Write a C program to find sum of n values a_i , i = 1 to n, using pointers instead of arrays. * Write a C program to count number of words in a given text by representing text string as pointer instead of array.	3 + 6
12.	* Remove all duplicates from an ordered array and contract the array accordingly.	1 + 2
13.	Explain: Numerical methods to find sqrt(n). * Given a number n >= 0, develop an algorithm and write a program in C to compute square root of a given non-negative number (n >= 0) by Divide-and-Conquer method * Write a program in C using an improved algorithm to compute Square Root using Newton's method and other methods.	2 + 4
14.	* Write a C program to find Maximum and Minimum values in a given array (or List) of values. Also write the C program using pointers instead of array.	1 + 2
	Total Hours	24 + 48

5. **Text Book**:

   1. Brian W. Kernigham, Dennis Ritchie, "The C Programming Language", Pearson (2015)
   2. R. G. Dromey, "How to Solve it by Computer", Pearson (2013)

6. **Reference Books**:
   - Yeshvant Kanetkar, "Let Us C", BPB Publication (2017)

   Webliography

   - https://www.w3resource.com/c-programming-exercises/
   - https://www.codechef.com/
   - https://www.learn-c.org/
   - https://www.prep.youth4work.com/practice-tests/c-programming-test
   - https://www.indiabix.com/online-test/c-programming-test/
   - https://www.hackerrank.com/c-programming-test-1/
   - https://www.mycplus.com/featured-articles/programming-contests-and-challenges/
   - https://www.hackerearth.com/challenges/
   - https://www.geeksforgeeks.org/category/competitive-programming/
   - https://www.techgig.com/challenge
   - https://www.freecodecamp.org/news/the-10-most-popular-coding-challenge-websites-of-2016-fb8a5672d22f/

7. **Course Outcome**:
   After studying this course, students will be able to write programs using iterative and recursive algorithms for various basic tasks such as to carry out the following tasks:
   - Sum and average of a given sequence of numbers using an array (a Pointer) or a List
   - Sum of infinite series, such as for trigonometric functions, etc
   - Using numerical methods to find Square root of a number
   - Prime numbers, prime divisors of n
   - GCD of given integers
   - Find n! and first n Fibonacci numbers using iterative and recursive algorithms
   - Find maximum and minimum in a given sequence on n numbers.
   - Remove duplicate values in an array.
   - Using Pointers, exchange 2 values
   - Improved algorithm for x^n

8. **Active Learning Assignment**
   - Simulate a simple dictionary. Assume each character contains at least 10 vocabularies. Create an index page for all characters. Retrieve the word using index value. Assume that the index characters are from a to z.
   - Design a simple search engine to display the possible websites upon entering a search query. Use suitable data structure for storage and retrieval

9. **Additional Exercises:**
   1. Given two variables of integer type a and b, exchange their values without using a third temporary variable.
   2. Write a program to print out n values of the following sequence: 1 – 1 1 – 1 1 – 1 …
   3. Write a program to compute the sum of the first n terms (n >= 1) of the following series: s = 1 – 3 + 5 – 7 + 9 – 11 + …
   4. For a given x and n, design an algorithm to compute x^n / n! .

5. Design an algorithm to compute ${}^{n}C_{r}$ = n! / [r! (n – r)!] for a given value of n and r.
    a. The exponential growth constant e is characterized by the infinite series: e = 1 / 0! + 1 / 1! + 1 / 2! + 1 / 3! + 1 / 4! + …
6. Design an algorithm to check whether two given integers are consecutive numbers of Fibonacci sequence or not.
7. Design an algorithm that computes (a) counts of number of digits, and (b) sum of digits in an integer.
8. Design an algorithm that reads in a set of n single digits and convert them into a single decimal integer. For example, the algorithm should convert a sequence of 6 digits (3, 2, 6, 4, 8, 5) to the integer 326485.
9. Design an algorithm that converts a given binary number to (a) Octal number and (b) Hexadecimal number.
10. Design an algorithm that accepts as input a decimal fraction and converts it into a corresponding binary fraction of a fixed accuracy. (For example $0.625_{10} = 0.101_{2} = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$)
11. Given that all ASCII codes are less than 128, design an algorithm that reads a given set of data and decides whether or not it may contain decimal data.
12. Design an algorithm that finds an integer whose square is closest to but greater than the integer number input as data.
13. For the integers in the range 1 to n (take n = 100), find the number that has the most divisors.
14. It is well known that adjacent Fibonacci numbers do not share a common divisor greater than 1 (they are relatively prime). Design an algorithm that tests this observation for the first n integers.
15. A perfect number is one whose divisors add up to the number. Design and implement an algorithm that determines all perfect numbers between 1 and 500. (Ex: 6 = 1 x 2 x 3; 6 = 1 + 2 + 3).
16. It is possible to compute n! In O (log2n) steps. Develop such an algorithm for computing n!.
17. Design an algorithm that rearranges the elements of an array such that all those numbers originally stored at odd suffixes are placed before those stored at even suffixes. For example, the input array 1, 2, 3, 4, 5, 6, 7, 8 will be transformed to 1, 3, 5, 7, 2, 4, 6, 8.
18. Generate Following Pattern.

1

```
N= 4
1
1 2
1 2 3
1 2 3 4
```

2.

```
N= 4
A
A B
A B C
A B C D
```

3.

```
N= 4
      A
    A B A
  A B C B A
A B C D C B A
```

4.

```
n= 4
          0
        1 0 1
      2 1 0 1 2
    3 2 1 0 1 2 3
```

5.

```
N= 4
      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

6.

```
N= 4
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

7.

```
N= 5
        +
      - -
    * * *
  + + + +
- - - - -
```

8.

```
N= 5
*                   *
* *               * *
* * *           * * *
* * * *   *   * * * *
* * * * * * * * * * *
```

9.

```
n= 4
      *
    *   *
  *   *   *
*   *   *   *
  *   *   *
    *   *
      *
```

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Oriented Programming in JAVA (OOPJ)
Subject Code: 619401

With effective from academic year 2020-21

1. **Learning Objectives:**
   - To develop proficiency in creating console based applications using the Java Programming Language.
   - To interpret the concepts of object oriented Programming Language and easily use these concepts in Java program.
   - To develop application using java.util package (Java data structure).
   - To implement multi-threaded applications using the Java Programming Language.
   - To implement application including different file operations.

2. **Prerequisites:** Knowledge of the C programming language and Object Oriented Concept

3. **Contents:**

Unit	Chapter Details	Weightage Percentage
Unit I	**Introduction to Java** **Java Fundamentals**: The Origins of Java,Java's Lineage: C and C++, How Java Impacted the Internet ,Java's Magic: The Bytecode, Moving Beyond Applets, The Java Buzzwords, Object-Oriented Programming(Encapsulation, Polymorphism, Inheritance), Obtaining the Java Development Kit,A First Simple Program, Handling Syntax Errors A Second Simple Program,Another Data Type, Two Control Statements (The if Statement and The for Loop), Create Blocks of Code, Semicolons and Positioning, Indentation Practices, The Java Keywords, Identifiers in Java, The Java Class Libraries Try This 1-1: Converting Gallons to Liters, Try This 1-2: Improving the Gallons-to-Liters Converter **Introducing Data Types and Operators:** Why Data Types Are Important, Java's Primitive Types(Integers, Floating-Point Types, Characters), The Boolean Type, ,Literals (Hexadecimal, Octal and Binary Literals, Character Escape Sequences, String Literals), A Closer Look at Variables(Initializing a Variable, Dynamic Initialization), The Scope and Lifetime of Variables, Operators, Arithmetic Operators (Increment and Decrement), Relational and Logical Operators, Short-Circuit Logical Operators, The Assignment Operator, Shorthand Assignments, Type Conversion in Assignments, Casting Incompatible Types, Operator Precedence, Expressions (Type Conversion in Expressions, Spacing and Parentheses) Try This 2-1: How Far Away Is the Lightning? Try This 2-2: Display a Truth Table for the Logical Operators **Program Control Statements:** Input Characters from the Keyboard, The if Statement, Nested ifs, The if-else-if Ladder, The switch	**15%**

	Statement, Nested switch Statements, The for Loop, Some Variations on the for Loop, Missing Pieces (The Infinite Loop), Loops with No Body, Declaring Loop Control Variables Inside the for Loop, The Enhanced for Loop, The while Loop, The do-while Loop, Use break to Exit a Loop, Use break as a Form of goto, Use continue, Nested Loops, Try This 3-1: Start Building a Java Help System, Try This 3-2: Improve the Java Help System, Try This 3-3: Finish the Java Help System,	
Unit II	**Class Fundamentals** **Introducing Classes, Objects, and Methods:** Class Fundamentals (The General Form of a Class, Defining a Class), How Objects Are Created, Reference Variables and Assignment, Methods (Adding a Method to the Vehicle Class), Returning from a Method, Returning a Value, Using Parameters (Adding a Parameterized Method to Vehicle), Constructors, Parameterized Constructors, Adding a Constructor to the Vehicle Class, The new Operator Revisited, Garbage Collection, The this Keyword, Try This 4-1: Creating a Help Class **A Closer Look at Methods and Classes:** Controlling Access to Class Members (Java's Access Modifiers), Pass Objects to Methods (How Arguments Are Passed), Returning Objects, Method Overloading, Overloading Constructors, Recursion, Understanding static (Static Blocks), Introducing Nested and Inner Classes, Varargs: Variable-Length Arguments (Varargs Basics, Overloading Varargs Methods, Varargs and Ambiguity) Try This 6-1: Improving the Queue Class, Try This 6-2: Overloading the Queue Constructor, Try This 6-3: The Quicksort,	**25%**
Unit III	**Inheritance Basics** **Inheritance:** Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Superclass Constructors, Using super to Access Superclass Members, Creating a Multilevel Hierarchy, When Are Constructors Executed?, Superclass References and Subclass Objects, Method Overriding, Overridden Methods Support Polymorphism, Why Overridden Methods? (Applying Method Overriding to TwoDShape), Using Abstract Classes, Using final, (final Prevents Overriding, final Prevents Inheritance, Using final with Data Members), The Object Class	**25%**

	Try This 7-1: Extending the Vehicle Class, **Packages:** Packages (Defining a Package, Finding Packages and CLASSPATH, A Short Package Example), Packages and Member Access (A Package Access Example), Understanding Protected Members, Importing Packages, Java's Class Library Is Contained in Packages, **Interfaces:** Interfaces, Implementing Interfaces, Using Interface References, Variables in Interfaces, Interfaces Can Be Extended, Default Interface Methods (Default Method Fundamentals, A More Practical Example of a Default Method, Multiple Inheritance Issues), Use static Methods in an Interface, Private Interface Methods, Final Thoughts on Packages and Interfaces, Try This 8-1: Creating a Queue Interface, **Exception Handling:** The Exception Hierarchy, Exception Handling Fundamentals (Using try and catch, A Simple Exception Example), The Consequences of an Uncaught Exception (Exceptions Enable You to Handle Errors Gracefully), Using Multiple catch Statements, Catching Subclass Exceptions, Try Blocks Can Be Nested, Throwing an Exception (Rethrowing an Exception), A Closer Look at Throwable, Using finally, Using throws, Three Additional Exception Features, Java's Built-in Exceptions, Creating Exception Subclasses, Try This 9-1: Adding Exceptions to the Queue Class	
Unit IV	**Autoboxing, java.lang package** Autoboxing, Type Wrappers, Autoboxing Fundamentals, Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions **Java.lang package (String, String Buffer, Comparable interface)** **The collection Framework:** Introduction, Collection framework (Collection interface, list interface, set interface, sorted set interface), The collection class, Array list and Link list classes (maintaining the capacity and the link list class **),** iterating elements of collection (the list iterator interface), hash set and tree set classes,	**15%**
Unit V	**Multi-Threading, Generic, I/O** **Multithreaded Programming** Multithreading Fundamentals, The Thread Class and Runnable Interface, Creating a Thread, (One Improvement and Two Simple Variations), Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, Using Synchronized Methods, The synchronized Statement	**20%**

	Try This 11-1: Extending Thread, Try This 11-2: Using the Main Thread, **Generic:** Generics Fundamentals, A Simple Generics Example, A Generic Class with Two Type Parameters, The General Form of a Generic Class), **Using I/O:** Java's I/O Is Built upon Streams, Byte Streams and Character Streams, The Byte Stream Classes, The Character Stream Classes, The Predefined Streams, Using the Byte Streams (Reading Console Input, Writing Console Output), Reading and Writing Files Using Byte Streams (Inputting from a File, Writing to a File), Automatically Closing a File, Reading and Writing Binary Data,	

Notes: Desirable:

1. java.util package and scanner class, formatter class, chapter 13 of Pravin Jain, "The class of Java" Pearson Education, (2010).
2. Working with Maps, map classes, The comparator interface
3. Thread communication using notify() , wait(0 and notifyall(), Suspending, resuming and stopping threads

4. Text Book:

1) Herbert Schildt,Java™: A Beginner's Guide, 7th Edition
2) Hari Mohan Pandey, JAVA Programming, 978-8131733110, Pearson

5. Reference Books:

1. Pravin Jain, "The class of Java" Pearson Education, (2010).
2. Paul J. Deitel, Harvey M. Deitel, Java SE8 for Programmers, ISBN: 9789332539068, Pearson
3. Cay S Horstmann, Gary Cornell, "Core Java 2, Volume 1 – Fundamentals", Pearson Education (8th edition – 2008).
4. Ivor Horton's "Beginning Java 2" JDK 5 Edition, Wiley Computer Publishing, (2007).
5. Ken Arnold, James Gosling, David Holmes, "The Java Programming Language", Addison-Wesley Pearson Education (4th Edition – 2005).
6. Raj Kumar Buyya, S. Thamarai Selvi, & Xing Chen Chu, "Object-Oriented Programming with Java: Essentials & Applications", Tata McGraw Hill
7. Cay Horstmann, "Big Java", Wiley Computer publishing (2nd edition – 2006).
8. Sharan Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser, Mark Hoeber "The Java Tutorial", Addison-Wesley Pearson Education(4th Edition),
9. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, "The Java Langauge

6. Webliography:

1. Oracle Java Documentation available for online reference at http://java.sun.com/docs/books/tutorial/index.html

2. Java SE Specifications available at https://docs.oracle.com/javase/specs/

Chapter wise Coverage from Main Reference Book(s):

Unit No.	Text Books	Topics/Subtopics
I	1	Chapter 1,2,3
II	1	Chapter 4,5,6
III	1	Chapter 7,8,9
IV	1	Chapter 12 ((pages 430 to 436) String (pages 158-164), StringBuffer(page 163, javadoc/api), Comparable interface (page: 464,466, javadoc/api)
IV	2	Chapter 18 (18.1 to 18.9 pages 534 to 566)
V	1	Chapter 11,13 (pages 445 to 453) ,10 (pages 331 to 351)

7. Accomplishments of the student after completing the course:

- Ability to create appropriate classes using the Java Programming Language to solve a problem using object oriented Approach.
- Ability to write console based applications in the Java Programming Language.
- Ability to develop to multi-threaded applications using the Java Programming Language
- Ability to develop file handling Java application.

8. Active Learning Assignment

- Consider any small application e.g. Library Management System having few transactions like Issue, Return, Renew and Report. Develop Java object oriented application programs and Present the application developed

9. Laboratory Exercises

A. List of Practical Related to JAVA:

General Instructions for Faculty Members/Lab Instructors:

- Use of integrated development environment (IDE) software application is restricted. All the programs mentioned in this list are to be performed using "Notepad or another text editor", JRE environment: Linux or Windows.
- Java is one of the most widely used programming languages in the field of Enterprise applications, Web Applications, Mobile Applications, Embedded systems, web server and applications server , networking applications , security functionality and environment and scientific applications because of its simplicity, modular programming and nice design etc.
- Students should be exposed to best programming practices for the given development environment.

1	Install the JDK (Download the JDK and install it.) • Set path of the jdk/bin directory. • Create the java program • Compile and run the java program Write a simple "Hello World" java program, compilation, debugging, executing using java compiler and interpreter.
2	Write a program to pass Starting and Ending limit and print all prime numbers and Fibonacci numbers between this range.
3	Write a java program to check whether number is palindrome or not. Input: 528 Output: It is not palindrome number Input: 545 Output: It is not palindrome number
4	Write a java program to print value of x^n. Input: x=5 Input: n=3 Output: 125
5	Write a java program to check Armstrong number. Input: 153 Output: Armstrong number Input: 22 Output: not Armstrong number
6	Write a program in Java to find minimum of three numbers using conditional operator.
7	Write a java program which should display maximum number of given 4 numbers.
8	Write a program in Java to multiply two matrix. Declare a class Matrix where 2D array is declared as instance variable and array should be initialized, within class.
9	Write a java program to create a class "Matrix" that would contain integer values having varied Numbers of columns for each row. Print row-wise sum of the integer values for each row.
10	Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below: (Suppose we enter 3 names then output should be as follows).. Number of arguments = 3 1.: First Student Name is = Arun 2.: Second Student Name is = Hiren 3.Third Student Name is = Hitesh
11	Write a Java application to count and display frequency of letters and digits from the String given by user as command-line argument.
12	Create a class "Student" that would contain enrollment No, name, and gender and marks as instance variables and count as static variable which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables.

	Also demonstrate constructor chaining. Create objects of class "Student" and displays all values of objects.
13	Write a program in Java to demonstrate use of **this** keyword. Check whether **this** can access the Static variables of the class or not. [Refer class student in Q12 to perform the task]
14	Create a class "Rectangle" that would contain length and width as an instance variable and count as a static variable. Define constructors **[constructor overloading (default, parameterized and copy)]** to initialize variables of objects. Define methods to find area and to display variables' value of objects which are created. [Note: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block]
15	Write a java program static block which will be executed before main ( ) method in a class.
16	Write programs in Java to use Wrapper class of each primitive data types.
17	Write a class "circle" with radius as data member and count the number of instances created using default constructor only. [**Constructor Chaining]**
18	Create a class "Vehicle" with instance variable vehicle_type. Inherit the class in a class called "Car" with instance model_type, company name etc. display the information of the vehicle by defining the display() in both super and sub class **[Method Overriding]**
19	Create a class "Account" containing accountNo, and balance as an instance variable. Derive the Account class into two classes named "Savings" and "Current". The "Savings" class should contain instance variable named interestRate, and the "Current" class should contain instance variable called overdraftLimit. Define appropriate methods for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account. [Ensure that the Account class cannot be instantiated.]
20	Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values. [refer class Account and sub classes savingAccount and CurrentAccount in Q 19 for this task]
21	Write a program in Java to demonstrate the use of **'final'** keyword in the field declaration. How it is accessed using the objects.
22	Write a java program to illustrates how to access a hidden variable. Class **A** declares a static variable **x**. The class **B** extends **A** and declares an instance variable **x**. **display ( )** method in **B** displays both of these variables.
23	Describe **abstract** class called **Shape** which has three subclasses say **Triangle**, **Rectangle**, **and Circle**. Define one method **area ()** in the abstract class and override this **area ()** in these three subclasses to calculate for specific object i.e. area () of Triangle subclass should calculate area of triangle etc. Same for **Rectangle** and **Circle**

24	Write a java program to implement an interface called Exam with a method Pass (int mark) that returns a boolean. Write another interface called Classify with a method Division (int average) which returns a String. Write a class called Result which implements both Exam and Classify. The Pass method should return true if the mark is greater than or equal to 50 else false. The Division method must return "First" when the parameter average is 60 or more, "Second" when average is 50 or more but below 60, "No division" when average is less than 50.
25	Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students.
26	Define a class A in package apack. In class A, three variables are defined of access modifiers protected, private and public. Define class B in package bpack which extends A and write display method which accesses variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and class display method for both these objects.
27	Write a java program to implement Generic class Number_1 for both data type int and float in java.
28	Write a java program to accept string to check whether it is in Upper or Lower case. After checking, case will be reversed.
29	Write a java program to use important methods of String class.
30	Write a program in Java to demonstrate use of final class, final variable and final method
31	Write a program in Java to develop user defined exception for 'Divide by Zero' error
32	Write a program in Java to demonstrate throw, throws, finally, multiple try block and multiple catch exception.
33	Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter.
34	Write a program to write at least 10 objects of the Circle class in a File and to perform basic operations: adding, retrieving, updating, removing elements**.**
35	Write a program for Java Generics class for Sorting operations: 1. Sorting a list according to natural ordering of elements 2. Reversing sort order 3. Sorting a list whose elements of a custom type 4. Sorting a list using a Comparator. [desirable]
36	Write a program in Java to create, write, modify, read operations on a Text file.
37	Write a java program to illustrate use of standard input stream to read the user input.
38	Write a java program to checks the existence of a specified file.
39	Write a java program to create a file to the specified location.
40	Write a java program to demonstrate the way contents are read from a file.
42	Write a java program to count the availability of text lines in the particular file. A file is read before counting lines of a particular file.

43	Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, palindromes).
44	Write a generic method to exchange the positions of two different elements in an array.
	Thread Programming
1	Write a program to implement the concept of threading by extending "Thread" Class.
2	Write a program to implement the concept of threading by implementing "Runnable" Interface.
3.	Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds.
4.	Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 50.
5.	Write java program that create and runs following threads: i) print "A" 20 times ii) print "B" 30 times iii) print "C" 15 times
6.	Write a program in Java to demonstrate use of synchronization of threads when multiple threads are trying to update common variable for "Account" class.
	The collection Framework
1.	Develop a program to create Linked List for "Student" class objects references. "Student" class has std_id, std_name, Array of marks, total_ marks. Calculate total_marks for all students of Linked List. Display Linked List and also display a particular student based on student name as a command line argument.
2.	Develop a program to create Array List for "Employee" class objects references. Employee class has emp_code, emp_name, basic_sal, gross_ sal. Calculate gross_sal for all employees of Array List. Display Array List and also insert an employee object reference in a particular position (input) in Array List. Gross_sal=basic_sal+20% of basic_sal (MA)+30% of basic_sal(HRA)
3.	Develop a program to create Hash Map for "Customer" class objects references. Customer class has Bill_no, cust_mobile_no, Array of item_name, Array of item_unit_price, Array of item_count, total_price. Calculate total_price for all customers of Hash Map. Display Hash Map and also search particular customer's bill based on customer mobile no. Key is mobile no.
4.	Sort "Student" Linked List (mentioned in Q:1) based on std_name using "Comparator" interface.

Reference: To learn Java language (Online tutorials):

1. https://docs.oracle.com/en/java
2. http://www.learnjavaonline.org/
3. http://java.sun.com/docs/books/tutorial/index.html

http://www.javaworld.com

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Basic Mathematics (BM)
Subject Code: 619403

With effective from academic year 2020-21

1. Objective

- The objective of this course is to present the foundations of many basic mathematical topics used in Computer Science including RDBMS, Data Structures, Analysis of Algorithms, Theory of Computation, Cryptography, Artificial Intelligence, Statistics and others.
- This course will enhance the student's ability to think logically and mathematically.

2. Prerequisites: School level Mathematics; Binary number system

3. Contents:

Sr. No.	Topics *(Proof of Theorems not required)*	Weightage Percentage
1	**Set Theory & Fundamentals** Set Theory: Basic Concepts of Set Theory: Definition, Two Methods to Describe (Represent) Sets; Sets and Subsets, Operations on Sets, Algebraic Properties of Set Operations, The Addition Principle, Sequences: Introduction, Characteristic Functions, Strings and Regular Expressions Matrices: Representation of a Matrix; Equality of Matrices; Special Matrices: Rectangular / Square Matrices, Null (Zero) Matrix, Unit Matrix, Diagonal Matrices, Sum and Difference of 2 Matrices; Multiplication of 2 matrices; Transpose of a Matrix, Symmetric Matrices, Boolean Matrix Operations	20%
2	**Mathematical Logic** Introduction, Propositions and Logical Operations, Truth Tables; DeMorgans' Laws; Conditional Statements, Methods of Proof, Mathematical Induction, Mathematical Statements	20%
3	**Integers and Counting** **Integers: Properties** of Integers; Prime Number; Greatest Common Divisor (GCD); Relative Prime; Least Common Multiple (LCM); Representation of Integers in Computer; Decimal, Binary, Octal, and Hexadecimal Representation **Counting :** Permutations, Combinations, Pigeonhole Principle, Elements of Probability, Recurrence Relations	15%
4	**Relations and Functions** Relations: Definition, Binary Relation, Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection, and Complement Operations on Relations, Properties of Binary Relations in a Set: Reflexive, Symmetric, Transitive, Antisymmetric Relations, Relation Matrix and Graph of a Relation; Partition and Covering of a Set, Equivalence Relation, Equivalence Classes, Compatibility Relation, Maximum Compatibility Block, Composite Relation, Converse of a Relation, Transitive Closure of a Relation R in Set X Functions: Introduction & Definition, Argument. Co-domain, Range,	25%

	Image, Value of a Function; Examples, Peano's Successor Function; Onto (surjective), Into, One-to-one (injective), Many- to-one, Bijective (one-to-one and onto); examples; Composition of Functions, examples; Inverse Function, Identity Map, Condition of a Function to be Invertible, examples; Inverse of Composite Functions, Properties of Composition of Functions; Binary and n-ary Operations as Mappings (functions), Properties of Binary Operations; Characteristic Function of a Set; Properties, examples; Hashing Functions: Division Method, and Mid-square Method, examples; *Proofs of Theorems not required*	
5	**Graphs and Trees** **Graphs**: Introduction, Definition; Initial & Terminal Nodes, Adjacent Nodes; Directed Edge, Undirected Edge, Directed Graph (Digraph), Undirected Graph, Mixed Graph; Loop (Sling); Distinct Edges, Parallel Edges; Multi-graph, Simple Graph; Weighted Graph; Isolated Nodes, Null Graph; Isomorphic Graphs; In-degree, Out-degree, Total-degree; Subgraphs; Reflexive, Symmetric, Transitive Digraphs; Paths, Length of Path of a Graph; Simple Path (Edge Simple), Elementary Path (Node Simple), Cycle (Circuit), Simple Cycle, Elementary Cycle; Path of Minimum Length (Geodesic), Distance between Two Nodes, Triangle Inequality; Reachability, Reachable Set of a Node, Connected Graphs: Strongly, Unilaterally, Weakly Connected Graphs & Components; **Trees**: Introduction, Definition, Root, Branch Nodes, Leaf (Terminal Node); Different Representations of Trees; Forests, Subtrees; M-ary Tree, Full or Complete M-ary Tree; Binary Tree, Full (Complete) Binary Tree; Conversion of M-ary Tree to Binary Tree; Traversal of Binary Tree: Pre-order, In-order, and Post-order Traversal;	20%

Note : *Proofs of Theorems not required*

4. Text Book:

1. J. P. Tremblay and R.Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw-Hill (2010) – only for Unit-5 (Graphs & Trees).
2. Bernard Kolmann & others, "Discrete Mathematical Structure", Pearson Education, Sixth Edition

5. Reference Books:

1. K. H. Rosen, "Discrete Mathematics and its applications", Tata McGraw-Hill, 6th edition,
2. D. S. Malik & M. K. Sen, "Discrete Mathematics", Cengage Learning (2004)
3. Edgar G. Goodaire, Michael M. Parmenter. "Discrete Mathematics with Graph Theory", PHI
4. Ralph P Grimaldi & B V Ramana, "Discreet and Combinatorial mathematics: An Applied Introduction", Pearson Education, 5th Edition (2018)
5. J. P. Tremblay and W. K. Grassman. "Logic and Discrete Mathematics", Pearson Education

6. Chapter-wise Coverage from the Text Book:

Unit No.	Text Books	Topics/Subtopics
I	Book-2	Chapter 1- 1.1 to 1.3 , 1.5
2	Book-2	Chapter 2 - 2.1 to 2.5
3	Book-2	Chapter 1 - 1.4, Chapter 3 - 3.1 to 3.5
4	Book-1	Chapter 2 - 2-3 (2-3.1 to2-3.7), 2-4 (2-4.1 to 2-4.6)
5	Book-1	Chapter 5 - 5-1 (5-1.1 to 5-1.3), 5-1.4, 5-2

7. Accomplishment of Students after Completing the Course:

Students will be able to understand and apply the concepts of sets, logic, cross product of sets and relation, functions, matrices, and basic algorithms related with binary tree and graphs.

8. Active Learning Assignment

Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work for matrix manipulation problem, which includes

- Introduction
- Flow chart
- Pseudo code
- Explanation

9. Laboratory Exercises

Write programs in C language for the following tasks.

Set Theory:

- Set Theory
  (a) Start with a NULL set and add elements one-by-one: Use different ways of implementing sets and understand the pros and cons of each of these methods
  (b) Given an element value, check whether it is a member of the set or not
  (c) Find out the number of elements of a given set.
  (d) Complement of a set; Union, Intersection
  (e) Test whether a given set X is a subset of the set A or not.
  (f) Test whether two given sets are equal or not
  (g) Difference and Symmetric Difference of two sets
- Create a set. Make sure that addition of elements does not accept any duplicate element. Assume that all elements of the set will be a non-negative integer < 64[0, 63].
  Hint: Create an array of size 64. Store the element x in index x.
  - Create 2 set A and B of size n_1 and n_2 . Print sets A and B.
  - Find ~A, ~B, A ∪ B, A ∩ B, A – B, B – A, and print the size of each derived set.
  - Find whether an element x ε A; x ε B; Find whether A ⊆ B; B ⊆ A; etc.

Desirable

- Assume that all elements of the set will be an uppercase letter ['A', 'Z'].
  Hint: Create an array of size 26. Store the element x in index x – 'A'. (Note: Try also with lowercase letters as elements of the set.)
  i. Create 2 set A and B of size n_1 and n_2 . Print sets A and B.
  ii. Find ~A, ~B, A ∪ B, A ∩ B, A – B, B – A, and print the size of each derived set.
  iii. Find whether an element x ε A; x ε B; Find whether A ⊆ B; B ⊆ A; etc.

Propositional Logic:

1. Consider 2 symbols: a and b. Represent various propositions using symbols a and b, such as ~a; ~b; (a ^ b); (a v b); (a => b); (b => a); (a <=> b). Determine the Truth value for each of these propositions for the following options:
   i. a is True and b is True
   ii. a is True and b is False
   iii. a is False and b is True
   iv. a is False and b is False
2. Verify DeMorgan's Laws: Hint: Find the Truth value of the LHS and RHS for 4 cases as mentioned above and compare.
   i. ~(a ^ b) = ~a v ~b
   ii. ~(a v b) = ~a ^ ~b

Matrices:

1. Write a function to Create a Matrix of size m x n, and another function to Print a Matrix of size m x n.
2. Create 2 matrices A and B of size (m x n). Find (A + B) and (A – B).
3. Find Transpose of matrix A and of matrix B. Find ($A^T + B^T$) and ($A^T – B^T$). Check whether ($A^T + B^T$) = $(A + B)^T$ and ($A^T – B^T$) = $(A – B)^T$.
4. Add matrix A and null matrix. Subtract null matrix from matrix A.
5. Create a unit matrix of size (n x n) and a unit matrix of size (n x n). Multiply matrix A with a unit matrix.
6. Create matrix A of size (m x n) and matrix B of size (n x p). Multiply matrix A and matrix B to get matrix C of size (m x p).
7. Create a symmetric matrix A. Find matrix A^T . Check whether A = A^T ?
8. Evaluate Scalar Product of a Matrix A: For example, k A, where k is a constant (number)
9. Take as input two matrices, A & B and print (A * B) and (B * A). First check which ones out of (A * B) and (A * B) are possible to compute.

Integers:

- Given a positive integer, find its divisors. Example: Divisors of 36 are 2, 3, 4, 6, 9, 12, 18.
- Given a positive integer n, represent n as product of its divisors. Example: 36 = 2 x 2 x 3 x 3.
- Given a positive integer, find whether it is a prime number or a composite number. Write an efficient algorithm.
- Given two positive integers, m and n, find whether they are relative prime numbers or not.
- Given two positive integers, a and b, find Least Common Multiples (LCM) of a and b.
- Given two positive integers a and b, find GCD (Greatest Common Divisor) of a and b.

- Given a positive integer n, convert it into a corresponding binary number, octal number and hexadecimal number.
- Addition of two binary numbers.
- Subtraction of a smaller binary number from a larger binary number.

Desirable

- Multiplication of a binary number by (a) 2, (b) 4, (c) 8, (d) 16
- Integer division (only quotient will be the answer) of a binary number by (a) 2, (b) 4, (c) 8, (d) 16
- Convert a negative integer into a corresponding binary number.
- Convert a decimal number into a corresponding binary number.
- Subtraction of two binary numbers.
- Multiplication of a binary number by another binary number.

Relations:

1. Input: Two sets A and B and a set of ordered pairs (a, b) such that a ε A, and b ε B. Output: Matrix representation of the Relation.
2. Input: Matrix representation of a relation.
   Output: Whether the relation is (a) Reflexive, (b) Symmetric, (c) Transitive, (d) Anti-symmetric.
3. Input: An Equivalence Relation.
   Output: Equivalence Classes

Functions:

1. Examples of Inverse Function:
   a. Given a positive integer, determine whether it is factorial of some integer or not?
   b. Given a positive integer, find out whether it belongs to Fibonacci sequence or not?
2. Input: Two matrices: Matrix 'a' of size (m x n) and Matrix 'b' of size (n x m).
   1. Output: Find out whether Matrix 'a' is inverse of Matrix 'b' or not?
3. Write functions floor1(), ceiling1() and round1() and test them.

Graphs and Trees :

- Input: Adjacency Matrix of a graph.
  - Output: Print In-degrees and Out-degrees of all nodes
  - Output: Print a sub-graph by removing the 1st node
  - Output: For the 1st node, Print all the paths. Also determine whether there are cycles in the graph? If yes, print the cycles.
- Input: Adjacency Matrix of a binary tree
  - Output: Print whether it is a complete binary tree? Print nodes which do not have 2 children.
  - Output: Print the length of left sub-tree and the length of right sub-tree.
  - Output: Print the sequence of nodes to be traversed during
    (a) Pre-order traversal, (b) in-order traversal, and (c) Post-order traversal

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Relational Database Management Systems (RDBMS)
Subject Code: 619404

With effective from academic year 2020-21

1. **Learning Objectives:**
   - To understand the fundamental concepts of Database Management Systems.
   - To understand the concepts necessary for designing, using and implementing database systems and applications

2. **Prerequisites:** Basic knowledge of working with computers

3. **Course Contents:**

Unit	Course Content	Weightage Percentage
Unit I	**Introduction to Database System** **Database and Users:** Introduction (Basic Concepts: Data, Database, Database systems, Database Management Systems), Characteristics of Database Approach, Actors on Scene, Workers behind the Scene, Advantages of using the DBMS approach **Database System Concepts and Architecture:** Data Models, Schemas, Instances, the three schema architectures and data independence, Database Languages and interfaces, Database System environment, Centralized and client / Server Architecture for DBMS, Classifications of Database Management Systems	**15%**
Unit II	**Entity Relationship Diagram** Using high level conceptual data models for database design (Design Phases of database design), Entity types, Entity Sets, Attributes and keys, Relationship Types, Relationship sets, Roles and structural constraints, Weak entity Types, Refining the ER diagram for company Database, Entity Relationship Diagram Naming conventions Design issues, Example of other Notation: UML class diagram, Relationship types of degree higher than 2 Subclasses, Super Classes, Inheritance Specialization and Generalization Relational Database design by ER and EER to Relational Mapping, Mapping EER model construct to Relations	**25%**
Unit III	**Database Design** Informal Design Guidelines for Relational Schema, Functional Dependencies, Normal Forms based on Primary keys, General definitions of 1NF, 2NF and 3NF, Boyce-Codd Normal Forms (BCNF), Multi-valued Dependency and Fourth Normal Form	**20%**
Unit IV	**Transaction processing** Introduction to Transaction Processing Concepts: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, characterizing Schedules based on recoverability and Serializability **Relational Model concepts**: Relational Model concepts, Relational Model constraints and Relational Database Schemas	**20%**

Unit V (*)	**SQL Concepts :**Basics of SQL, DDL,DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions –numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. transaction control commands – Commit, Rollback, Savepoint	**10%**
Unit VI (**)	**PL/SQL Concepts** PL/SQL Block, Stored Procedures, Functions and Packages (Except Cursor Management)	**-**

(*): Only Higher order questions / application oriented questions to be asked in the theory exam from Unit V and VI.
(**): Only for practical examination.

Desirable:

i) Introduction to other databases NoSQL Databases (Mongodb, Cassandra, Redis etc.) , PostgreSQL, MySQL. Etc.
ii) Relational Algebra
iii) Implementing security in databases

4. Text Book:

1. Ramez Elmsari,Shamkant B Navathe, "Fundamentals of Database Systems", Pearson Education, 7th Edition
2. Ivan Bayross, SQL, PL/SQL the Programming Language of Oracle, 4th Edition, BPB Publications

5. Reference Books:

1. Silberschatz, Korth, Sudarshan, "Database System Concepts", McGraw Hill Publication. 5th Edition
2. S K Singh, "Database Systems : Concepts, Design and Applications", Pearson Education
3. Peter Rob, Carlos Coronel, "Database Systems : Design, Implementation and Management", Cengage Learning
4. C J Date, A Kannan, S Swaminathan, "An Introduction to Database Systems", Pearson Education, 8th Edition
5. Steve Suehring, Tim Converse, Joyce Park, PHP 6 and MySQL Bible, Wiley
6. Andrea Tarr, PHP and MySQL 24-Hour Trainer, Wiley

Web references:

1) https://docs.oracle.com/en/database/index.html
2) https://docs.oracle.com/database/121/SQLRF/toc.htm

6. Chapter wise Coverage from the Text Book:

Unit No.	Text Books	Topics/Subtopics
1	Book-I	Chapter 1 (1.1 to 1.6), Chapter 2

2	Book-I	Chapter 3, Chapter 4 (4.1 & 4.2), Chapter 9
3	Book-I	Chapter 14 (14.1 to 14.6)
4	Book-I	Chapter 20(20.1 to 20.5) Chapter 5 (5.1 & 5.2),
5	Book II	Chapter 7,8,9,10,11 (except cluster and snapshot)
6	Book II	

7. **Accomplishment of the student after completing the course**:
   - A student would be able to effectively squeeze the "real world" data into the relational data model of the database system and data retrieval
   - Clear understanding for the need of a database.
   - Understand the uses the database schema
   - Understand the need for normalization
   - Use different types of physical implementation of database

8. **Active Learning Assignment**
   Consider Leave Management process includes defining the leave types, assigning entitlements and calculating carry over leaves, employees applying for leaves, managers approving or rejecting the leave requests, importing the leave data into payroll for calculations etc.

   Prepare Database design presentation which includes:
   1. Purpose of database
   2. Find/ Identify Information required for application
   3. Conceptual Design
   4. Logical Design
   5. Physical Design

   Apply the concepts learned of conceptual design and logical database design

9. **Laboratory**

Practicals

Part I: RDBMS (Module Weightage: 100%)

Tools: Oracle 10g or above
Topics:

1	Data Types
2	Study of DDL Commands (Create , Alter , drop) Table: The Create Table Command, Creating a table from a table (with data, without data, with all columns, with selected columns), Drop Table, Alter Table, Renaming Tables
3	Study DML Commands (Select, insert, update, delete)
4	Constraints: Defining integrity constraints using create table and the alter table command, Dropping integrity constraints in the alter table command
5	Transaction Control statements: Commit, Rollback
6	Advanced Concepts: View, Index, Sequences, rowed, rownum, Default Value Concept
7	Join (Inner Join, Equi Joins, Self Join, Outer Joins)

8	Study subquery concepts
9	Set Operators
10	Study single row functions: String functions, Numeric Functions, Date Functions, Date Conversion Functions
11	Study aggregate / group functions, having
12	Sorting Data, Handling Null values (IS NULL), Like Clause
13	Basic concepts of PL/SQL
14	Create Triggers
15	Data dictionary
16	Procedure, function, package
17	**Desirable** : Security / privileges

SET 1

DEPARTMENT (dept_no, dept_name, location)

1. Create the Simple DEPARTMENT Table.
2. Display structure of department table.
3. Insert below records into Department Table

Dept_no	Dept_name	Location
10	Account	NY
20	HR	NY
30	Production	DL
40	Sales	NY
50	EDP	MU
60	TRG	
110	RND	AH

4. Display all records of Department table
5. Display all department belonging to location 'NY'
6. Display details of Department 10
7. List all department names starting with 'A'
8. List all departments whose number is between 1 and 100
9. Delete 'TRG' department
10. Change department name 'EDP' to 'IT'

SET 2

EMPLOYEE (emp_id, emp_name, birth_date, gender, dept_no, address, designation, salary, experience, email)
DEPARTMENT (dept_no, dept_name, location)

Do as directed:

1. Create the EMP Table with all necessary constraints such as
   In EMP TABLE: Employee id should be primary key, Department no should be Foreign key, employee age (birth_date) should be greater than 18 years, salary should be greater than zero, email should have (@ and dot) sign in address, designation of

employee can be "manager", "clerk", "leader", "analyst", "designer", "coder", "tester".

2. Create DEPT table with neccessary constraint such as
3. Department no should be primary key, department name should be unique.
4. After creation of above tables, modify Employee table by adding the constraints as
5. 'Male' or 'Female' in gender field and display the structure.
6. Insert proper data (at least 5 appropriate records) in all the tables.
7. Describe the structure of table created
8. List all records of each table in ascending order.
9. Delete the department whose loction is Ahmedabad.
10. Display female employee list
11. Display Departname wise employee Names
12. Find the names of the employee who has salary less than 5000 and greater than 2000.
13. Display the names and the designation of all female employee in descending order.
14. Display the names of all the employees who names starts with 'A' ends with 'A'.
15. Find the name of employee and salary for those who had obtain minimum salary.
16. Add 10% raise in salary of all employees whose department is 'IT'.
17. Count total number of employees of 'IT' department.
18. List all employees who born in the current month.
19. Print the record of employee and dept table as "Employee works in department 'MBA'.
20. List names of employees who are fresher's (less than 1 year of experience).
21. List department wise names of employees who has more than 5 years of experience.
22. Crete Sequence to generate department ID
23. List department having no employees

SET 3

STUDENT (rollno, name, class, birthdate)
COURSE (courseno, coursename, max_marks, pass_marks)
SC (rollno, courseno, marks)

1. Create the above three tables along with key constraints.
2. Write an Insert script for insertion of rows with substitution variables and insert appropriate data.
3. Add a constraint that the marks entered should strictly be between 0 and 100.
4. While creating SC table, composite key constraint was forgotten. Add the composite keynow.
5. Display details of student who takes 'Database Management System' course.
6. Display the names of students who have scored more than 70% in Computer Networksand have not failed in any subject.
7. Display the average marks obtained by each student.
8. Select all courses where passing marks are more than 30% of average maximum mark.
9. Display details of students who are born in 1980 or 1982.
10. Create a view that displays student courseno and its corresponding marks.

SET 4

Create the database COMPANY and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

EMPLOYEE (emp_id, emp_name, birth_date, gender, dept_no, address, designation, salary, experience, email)
DEPART (dept_no, dept_name, total_employees, location)
PROJECT (proj_id, type_of_project, status, start_date, emp_id)

Insert proper data (at least 5 appropriate records) in all the tables.

Do as directed:

1. Delete the department whose total number of employees less than 1.
2. Display the names and the designation of all female employee in descending order.
3. Display the names of all the employees who names starts with 'A' ends with 'A'.
4. Find the name of employee and salary for those who had obtain minimum salary.
5. Add 10% raise in salary of all employees whose department is 'CIVIL'.
6. Count total number of employees of 'MCA' department.
7. List all employees who born in the current month.
8. Print the record of employee and dept table as "Employee works in department 'CE'.
9. List names of employees who are fresher's(less than 1 year of experience).
10. List department wise names of employees who has more than 5 years of experience.
11. Write a function which will display total number of projects based on status (pass status as parameter).
12. Write a procedure that will display list of projects which is going to start today.
13. Write a trigger which do not allow insertion/updation/deletion into Project table if status type is 'pending'

SET 5

Create the database STUD and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

HOSTEL (HNO, HNAME, HADDR, TOTAL_CAPACITY, WARDEN)
ROOM (HNO, RNO, RTYPE, LOCATION, NO_OF_STUDENTS, STATUS)
CHARGES (HNO, RTYPE, CHARGES)
STUDENT (SID, SNAME, MOBILE-NO, GENDER, FACULTY, DEPT, CLASS, HNO, RNO)
FEES (SID, FDATE, FAMOUNT)

The STATUS field tells us whether the room is occupied or vacant. The charges represent the term fees to be paid half yearly. A student can pay either the annual fees at one time or the half yearly fees twice a year.
Insert proper data (at least 5 appropriate records) in all the tables.

Do as directed:

1. Display the total number of rooms that are presently vacant.
2. Display number of students of each faculty and department wise staying in each hostel.
3. Display hostels, which have at least one single-seated room.

4. Display the warden name and hostel address of students of Computer Science department.
5. Display those hostel details where single seated or double-seated rooms are vacant.
6. Display details of hostels occupied by medical students.
7. Display hostels, which are totally occupied to its fullest capacity.
8. List details about students who are staying in the double-seated rooms of Chanakya Hostel.
9. Display the total number of students staying in each room type of each hostel.
10. Display details about students who have paid fees in the month of Nov. 2017.
11. For those hostels where total capacity is more than 300, display details of students studying in Science faculty.
12. Display hostel details where there are at least 10 vacant rooms.
13. Display details of students who have still not paid fees.
14. Display those hostels where single-seated room is the costliest.
15. Write a trigger which do not allow to insert or update student record if mobile_no length is less than 10 digits.
16. Write a PL/SQL block which will count total number of student's gender wise.
    Male Students: 999 students
    Female Students: 999 students

SET 6

Create the database HOSPITAL and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

DOCTOR (DNO, DNAME, SPECIALIZATION, CLINIC_ADDR)
MEDICINE (MNO, MNAME, TYPE, CONTENT, MANUFACTURER)
DISEASE (DISEASE_NAME, SYMPTOM1, SYMPTOM2, SYMPTOM3)
TREATMENT (TNO, DNO, DISEASE_NAME, MNO, DOSAGE, AVG_CURE_TIME)

Insert proper data (at least 5 appropriate records) in all the tables.

Do as directed:

1. Display records of each table in ascending order.
2. Count total number of doctors which has not given any treatment.
3. Display all Chennai doctors who treat cancer.
4. Remove disease "polio" from disease table as well as treatment table.
5. Delete all those treatment related to liver of Dr.Shah.
6. Create index on dno, Disease name in the treatment table.
7. Display details of doctors who treat migraines.
8. What is the maximum dosage of "penicillin" prescribe by the doctor for the treatment of any disease?
9. Display total number of disease treated by every doctor.
10. Which doctor have no treatment for "depression"?
11. Create a view which contains the treatment and doctors details. Make sure that no body is allowed to modify any detail in the view.
12. Write a PL/SQL block to print the following report (Symptoms wise print total number of medicine given)

Medicine	Symptom1	Symptom2	Symptom3	Total
M1	999	999	999	9999
M2	999	999	999	9999
M3	999	999	999	9999

13. Write a trigger which does not allow to insert or update treatment table if AVG_CURE_TIME is less than 1.

SET 7

Create the database SHOPPING and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

CUSTOMER (cno, cust_name, cust_phone, location,gender)
ITEM (itemno, itemname, color, weight, expire_date, price, shop_name)
CUST_ITEM (cno, itemno, quantity_purchased, date_purchase)

Insert proper data (at least 5 appropriate records) in all the tables.

Do as directed:

1. Delete the items whose price is more than 50000.
2. Find the names of the customer who is located in same location as that of other customer.
3. Display the names of items which is black, white & brown in color.
4. Display the names of all the items whose names lies between 'p' and's'.
5. Find the item which is having less weight.
6. Add one month more to those items whose item no =40.
7. Count total number of items which is going to expire in next month
8. List all customers whose phone number starts with '99'.
9. Display total value (qty*price) for all items.
10. List customer details who has purchased maximum number of items
11. Display total price item wise.
12. List name of items, customer details and qty purchased.
13. Write a PL/SQL procedure which will display records in the following format

Today's Date: _____ Shop name: _____

Item number	Item name	Expire date	Quantity	Price	Total Rs.

Grand Total Rs. _____

14. Write a trigger which do not allow insertion / updation / deletion of Item details on Sunday.

SET 8

Create the database THEATRE and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

SCREEN (SCREEN_ID, LOCATION, SEATING_CAP)
MOVIE (MOVIE_ID, MOVIE_NAME, DATE_OF_RELEASE)
CURRENT (SCREEN_ID, MOVIE_ID, DATE_OF_ARRIVAL, DATE_OF_CLOSURE)

Check Constraints:
Value of screen_id must start with letters 'S'.
Attribute location can be any one of 'FF', 'SF', or 'TF'.

Do as directed:

1. Get the name of movie which has run the longest in the multiplex so far.
2. Get the average duration of a movie on screen number 'S4'.
3. Get the details of movie that closed on date 24-november-2004.
4. Movie 'star wars III 'was released in the 7th week of 2005. Find out the date of its release considering that a movie releases only on Friday.
5. Get the full outer join of the relations screen and current.
6. Write a PL/SQL function which will count total number of day's horror movie last longer.
7. Write a PL/SQL procedure that will display movie which is going to release today.
8. Write a trigger which will not allow to insert/update in current table if Date_of_arrival is less than date_of_closure.

SET 9

Create the database EXAM and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

APPLICANT (AID, ANAME, ADDR, ABIRTH_DT)
ENTRANCE_TEST (ETID, ETNAME, MAX_SCORE, CUT_SCORE)
ETEST_CENTRE (ETCID, LOCATION, INCHARGE, CAPACITY)
ETEST_DETAILS (AID, ETID, ETCID, ETEST_DT, SCORE)

(This database is for a common entrance test which is being conducted at a number of centers and can be taken by an applicant on any day except holidays)

Do as directed:

1. Modify the APPLICANT table so that every applicant id has an 'A' before its value. E.g. if value is '1123', it should become 'A1123'.
   Display test center details where no tests were conducted.
   Display details about applicants who have the same score as that of Ajaykumar in 'ORACLE FUNDAMENTALS'.
2. Display details of applicants who appeared for all tests.
3. Display those tests where no applicant has failed.

4. Display details of entrance test centers which had full attendance between 1st Oct 15 and 15th Oct 16.
5. Display details of the applicants who scored more than the cut score in the tests they appeared in.
6. Display average and maximum score test wise of tests conducted at Mumbai.
7. Display the number of applicants who have appeared for each test, test center wise.
8. Display details about test centers where no tests have been conducted.
9. For tests, which have been conducted between 2-3-17 and 23-4-17, show details of the tests as well as the test centre.
10. How many applicants appeared in the 'ORACLE FUNDAMENTALS' test at Chennai in the month of February?
11. Display details about applicants who appeared for tests in the same month as the month in which they were born.
12. Display the details about APPLICANTS who have scored the highest in each test, test centre wise.
13. Design a read only view, which has details about applicants and the tests that he has appeared for.
14. Write a procedure which will print maximum score centre wise.
15. Write a procedure which will print details of entrance test.

Centre name: _____ candidate id: ____ date: ___ score: _____

---

15. Write a trigger which do not allow insertion / updation / deletion of Enterance test details on Sunday.

SET 10

Create the database BUS TRANSPORT and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

CATEGORY (CAT_CODE, CATDESC)
ROUTEMASTER (ROUTENO, ORIGIN, DESTINATION, FARE, DISTANCE, CAPACITY, DAY, CAT_CODE)
TICKETHEADER (TICKETNO, DATEOFISSUE, DATEOFTRAVEL, BOARDPLACE, ROUTENO)
TICKET DETAILS (TICKETNO, NAME, SEX, AGE, FARE)

ADD THE FOLLOWING CONSTRAINTS:

1. DELUXE, SUPERDELUXE, SUPERFAST AND NORMAL ARE THE CATDESC
2. ORIGIN AND DESTINATION CANNOT BE SAME,
3. CAPACITY SHOULD BE>0 AND <=60

Do as directed:

1. Display the total number of people traveled on each ticket group by ticket no 23.
2. Give the total collection of fare for each route.
3. Give the number of months between issue date and travel date of each ticket issued.
4. Count number of person boarding from the same place and same route.

5. Display count of person who has traveled in each category.
6. Write a trigger which allow to insert or update the bus capacity only greater than zero and less than 60.
7. Write a Procedure which will print tour details, a driver is going to take it. (pass route_no as parameter)

Route No: _____ Vehicle number: _____

Transport Details

Source	Destination	Start date	Total days	Capacity
Xxx	Xxx	Xxx	999	999

SET 11

Create the database BUS TRANSPORT and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

TRAIN _MASTER:

FIELD NAME	DATA TYPE	CONSTRAINTS
TRAIN NUMBER	VARCHAR2(6)	PRIMARY KEY AND LAST TWO CHARS SHOULD BE 'DN' OR 'UP'
TRAIN NAME	VARCHAR2(25)	NOT NULL
ARRIVAL TIME	DATE	NOT NULL
DEPARTURE TIME	DATE	NOT NULL
NO OF HOURS	NUMBER(5,2)	NOT NULL
SOURCE STATION	VARCHAR2(25)	NOT NULL
END STATION	VHARCHAR2(25)	NOT NULL

PASSENGER_DETAILS:

FIELD NAME	DATA TYPE	CONSTRAINTS
TICKET NUMBER	NUMBER(5)	
TRAIN NUMBER	VARCHAR2(6)	FOREIGN KEY REFERENCE TO TRAIN_MASTER THIS RELATED RECORD SHOULD BE DELETED IF MASTER RECORD IS DELETED.
SEAT NUMBER	NUMBER(2)	NOT NULL
PASSENGER NAME	VARCHAR2(35)	NOT NULL
AGE	NUMBER(2)	NOT NULL
GENDER	CHAR(1)	SHOULD BE 'M' FOR MALE OR 'F' FOR FEMALE
TRAVEL DATE	DATE	
CLASS	VARCHAR2(4)	SHOULD BE IN (IA, IIA, IIIA, IC, II)

TRAIN_SEAT_MASTER:

FIELD NAME	DATA TYPE	CONSTRAINTS
TRAIN NUMBER	VARCHAR2(6)	FOREIGN KEY REFERENCE TO TRAIN_MASTER THIS RELATED RECORD SHOULD BE DELETED IF MASTER RECORD IS DELETED.
CLASS	VARCHAR2(4)	SHOULD BE IN (IA, IIA, IIIA, IC, II)
TOTAL SEATS	NUMBER(2)	SHOULD BE >= 25 AND <= 90

TRAIN_DAY_MASTER:

FIELD NAME	DATA TYPE	CONSTRAINTS
TRAIN NUMBER	VARCHAR2(6)	FOREIGN KEY REFERENCE TO TRAIN_MASTER THIS RELATED RECORD SHOULD BE DELETED IF MASTER RECORD IS DELETED.
DAY	VARCHAR2(3)	VALUE SHOULD BE IN 'MON' … TO … 'SUN'

Do as directed:

1. Give all the train nanes starting from “Bombay” and going to “Ahmedabad” on Tuesday and Wednesday.
2. List all trains which is available on Sunday.
3. Give classwise seat availability on 10-June-2018 for train 9012DN.
4. List total seats classwise for train running on thrusday.
5. List train names which have no sleeper class.
6. List train number which run on Monday during 8:00: am to 1:00pm.
7. Write a procedure which will print all train details going from Baroda to Banglore.
8. Write a function which will print arrival time and departure time for a given train. (pass train no as a parameter)
9. Write a trigger which do not allow to insert or update passenger record if age is greater than 100.

SET 12

CUSTOMER(cid, fname, lname, city, country, phone)
ORDER (oid, oDate, oNumber, cid, oTotalAmount)

1. List the number of customers in each country. Only include countries with more than 100 customers.
2. List the number of customers in each country, except China, sorted high to low. Only include countries with 5 or more customers.
3. List all customers with average orders between Rs.5000 and Rs.6500.
4. Create a trigger that executes whenever country is updated in CUSTOMER table.

5. Create a function to return customer with maximum orders.
6. Create a procedure to display month names of dates of ORDER table. The month names should be unique.

SET 13

DISTRIBUTOR (dno, dname, daddress, dphone)
ITEM (itemno, itemname, colour, weight)
DIST_ITEM (dno, itemno, qty)

1. Add a column CONTACT_PERSON to the DISTRIBUTOR table with the not null constraint.
2. Create a view LONDON_DIST on DIST_ITEM which contains only those records where distributors are from London. Make sure that this condition is checked for every DML against this view.
3. Display the details of all those items that have never been supplied.
4. Delete all those items that have been supplied only once.
5. List the names of distributors who have an 'A' and also a 'B' somewhere in their names.
6. Count the number of items having the same colour but not having weight between 20 and 100.
7. Display all those distributors who have supplied more than 1000 parts of the same type.
8. Display the average weight of items of the same colour provided at least three items have That colour.
9. Display the position where a distributor name has an 'OH' in its spelling somewhere after the fourth character.
10. Count the number of distributors who have a phone connection and are supplying item number 'I100'.
11. Create a view on the tables in such a way that the view contains the distributor name, item name and the quantity supplied.
12. List the name, address and phone number of distributors who have the same three digits in their number as 'Mr. Talkative'.
13. List all distributor names who supply either item I1 or I7 or the quantity supplied is more than 100.
14. Display the data of the top three heaviest ITEMS.

SET 14

1. Write a PLSQL block which will print Employee list (Empno and Name)
   EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)
2. Write a function that returns total number of incomplete jobs, using table
   JOB (jobid, type_of_job, status)
3. Write a function which displays the number of items whose weight fall between a given ranges for a particular color using table ITEM (itemno, name, color, weight)
4. Write a procedure to display top five highest paid workers who are specialized in 'PAINTING" using table WORKER (workerid, name, wage_per_hour, specialized_in, manager_id)

SET 15

EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)
DEPT (deptno, deptnm)

Write a PL/SQL block (table above EMP-DEPT table) which takes as input Department name and displays all the employees of this department who has been working since last five years

SET 16

EMPMAST (empno, name, pfno, empbasic, deptno, designation)
DEPT (DNO, DNAME)

Rules: HRA = 15% of basic
DA = 50% of basic
Medical = 100
PF = 8.33%of basic

Print Salary slip. Design your own format

SET 17

Consider the Bank schema as
ACCOUNT (AC_NO, NAME, AC_TYPE, BALANCE_AMT, BALANCE_DATE)
TRANSACTION (AC_NO, DATE, TR_TYPE, AMOUNT, PREV_BALANCE, REMARK)

Note: 1. AC_type may be S for saving or C for current, 2. TR_type may be D for deposit or W for withdrawal.
a. Write a procedure to print the Bank Transaction details by passing from and to dates.

SET 18

BRANCH (branch_no, area, city)
MEMBERS (mno, name branch_no, salary, manager_no)
Note: Manager can be from one of the members.
1. Write a procedure which list the name of members who earns more than that of his managers.
2. Write a procedure which gives details of employee having maximum salary branch wise.

SET 19

Employee (eid, fname, lname, salary)

1. Use a Cursor for Loop inside a function to calculate and return total paid salary to all employees by the company.
2. Modify the function created above to become a procedure and display the total paid salary from the procedure itself. Instead of calculating for all employees, calculate only for those employees whose name starts from a character passed as parameter to the procedure and hence to the cursor.

SET 20

Consider the DUAL and data dictionary tables/views to solve the following Queries.

1. Find out the names of all the tables, views and constraints associated with current tables in the system.
2. Write a query to add 15 days to the current date.
3. Write a query to Add and subtract 5 months from the current month.
4. Find out the ASCII equivalent of character 'M'.
5. Find out the character equivalent of ASCII 67, 65 and 84.
6. Write a query to find the last day of the month.
7. Find out how many days are left in the current month.
8. Write a query to calculate the Date difference between current date and 20/05/2015.
9. Write a query to Calculate the number of months between current date and 03/03/2016.
10. Find out the second occurrence of 'or' from third position in the string 'corporate floor .
11. Find out log to the base 3 of 81.
12. Convert the string 'gujarat technological university' so that first character of each work is in capital.
13. Convert the string 'jack and jue' Into 'black and blue'.
14. Round off the date 27-July-2016 to the current year.
15. Find out the user name and user id off currently logged on user.

. **Part II: NoSQL Database (Desirable)**

Tools: MongoDB

1	Introduction, Installation
2	Create Database, Drop Database
3	Create Collection, show collection
4	Insert document, Query Document, Update document, delete document
5	Projection
6	Limiting rows
7	Export and Import

https://docs.mongodb.com/manual/mongo/

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Web Development Technology
Subject Code: 619405

With effective from academic year 2020-21

1. Learning Objectives:

- To understand Requirement Analysis, System Design, Quality Assurance, and Implementation.
- To understand various Software Development Life Cycle (SDLC) models and criteria to select appropriate model.
- To understand how to Analyze, Design, Build and Test software
- To understand Agile Methodology.

2. Prerequisites: HTML, CSS and Basics of MySQL

3. Contents:

Unit	Chapter Details	Weightage Percentage
Unit I (*)	**JavaScript Overview** Understanding JavaScript, uses of JavaScript, attaching external JavaScript, validating form fields using JavaScript, extending functionalities of form fields using JavaScript **JSON** Overview, Syntax, Data Types, Objects, Schema, Serializing into JSON, Parsing JSON	-
Unit II	**Introduction to PHP** **Introduction:** Basic HTML syntax, Basic PHP Syntax, using SFTP, Testing your script, Sending text to browser, Using the PHP Manual, Sending HTML to the browser, Adding comments to Scripts, Basic debugging steps **Variables:** What Are Variables?, Variable Syntax, Types of Variables, Variable Values, Understanding Quotation Marks **Form:** Creating a Simple Form, Choosing a Form Method, Receiving Form Data in PHP, Displaying Errors, Error Reporting, Manually Sending Data to a Page **Numbers:** Creating the Form, Performing Arithmetic, Formatting Numbers, Understanding Precedence, Incrementing and Decrementing a Number, Creating Random Numbers **Strings:** Creating the HTML Form, Concatenating Strings, Handling Newlines, HTML and PHP, Encoding and Decoding Strings, Finding Substrings, Replacing Parts of a String **Control Structures:** Creating the HTML Form, The if Conditional, Validation Functions, Using else, More Operators, Using elseif The Switch Conditional, The for Loop	-

Unit III	**Working with Arrays, Forms and Functions** **Using Arrays:** What Is an Array?, Creating an Array, Adding Items to an Array, Accessing Array Elements, Creating Multidimensional Arrays, Sorting Arrays, Transforming Between Strings and Arrays, Creating an Array from a Form **Creating Web Applications:** Creating Templates, Using External Files, Using Constants, Working with the Date and Time, Handling HTML, Forms with PHP, Revisited, Making Forms Sticky, Sending Email, Output Buffering, Manipulating HTTP Headers **Creating Functions:** Creating and Using Simple Functions, Creating and Calling Functions that Take Arguments, Setting Default Argument Values, Creating and Using Functions that Return a Value, Understanding Variable Scope	=
Unit IV	**Advanced PHP** **Cookies and Sessions:** What Are Cookies?, Creating Cookies, Reading from Cookies, Adding Parameters to a Cookie, Deleting a Cookie, What Are Sessions?, Creating a Session, Accessing Session Variables, Deleting a Session **Files and Directories**: File Permissions, Writing to Files, Locking Files, Reading from Files, Handling File Uploads, Navigating Directories, Creating Directories, Reading Files Incrementally	-
Unit V	**jQuery Basics** Exploring Fundamentals of jQuery, loading and using jQuery, describing call back functions, exploring jQuery Selectors, methods, manipulators, events and effects, exploring jQuery and AJAX	=

Note: Above content is provided for tutorial planning and practicals.

4. Text Book:

1. Jeremy McPeak, Paul Wilton, Beginning JavaScript Wrox Publication
2. Larry Ullman, PHP for Web, fifth Edition, Pearson
3. Adam Freeman Pro jQuery 2.0, Apress

5. Reference Books:

1) Julie C Meloni, "Sams Teach Yourself PHP, MySQL and Apache All in One" 4th
2) edition, Pearson Education
3) HTML5 Black Book: Covers CSS3, Javascript, XML, XHTML, Ajax, PHP and
4) Jquery, by Kogent Learning Solutions Inc.
5) Tim Converse and Joyce Park, PHP6 and MySQL Bible –Steve Suehring, Wiley India

6) Latest Edition.
7) Luke Welling, Laura Thomson , PHP and MySQL Web Development — Pearson
8) Beginning Ajax with PHP From Novice to Professional, By Lee BabinApress
9) Head First AJAX by Rebecca Riordan , O'Reilly Media
10) Head First PHP& MySQL by Lynn Beighley, Michael Morrison, O'Reilly Media
11) Head First jQuery by Ryan Benedetti and Ronan Cranley, O'Reilly Media
12) Learning jQuery By Jonathon chaffer and Karl Swedberg, O'Reilly Media

Webliography:

1. http://www.codecademy.com/learn
2. http://www.w3schools.com
3. https://developer.mozilla.org/en/learn/javascript
4. http://www.learn-javascript-tutorial.com/

1) Chapter wise Coverage from Main Reference Book(s):

Unit No.	Text Books	Topics/Subtopics
I	Book-1	Chapter 1,10,11,12
II	Book-2	Chapter: 1,2,3,4,5,6
III	Book-2	Chapter: 7,8,10
IV	Book-2	Chapter : 9,11
V	Book-3	Chapter : 5, 9, 10, 14, 15

7. Accomplishments of the student after completing the course:

- Ability to install and configure MySQL, Apache and PHP.
- Ability to develop web based applications using PHP interacting with database.
- Ability to set & access cookies & session variables, Upload a file in PHP.
- Ability to validate forms using AJAX
- Ability to integrate AJAX and JQUERY in PHP.

5. Laboratory Exercises

1. Write a PHP program to display current date and time and display Good Morning / Good Afternoon / Good Evening message according to current time.
2. Create a web page for user profile and execute a PHP file on submission of the form and display the information using PHP.
3. Write a PHP Program to perform following operation on Array where values in array are entered by user

a) Print the values of array.
b) Reverse an array.
c) merge two arrays in sorted manner.
d) add values of all elements of an array.

4. Write a PHP function

a) to print your name.

b) to print the size of a string. Pass string as an argument
c) to accept variable length arguments and display sum of all values and total number of arguments.

5. Write a PHP program to perform following string operations:
   a) print your name.
   b) print the size of a string. Pass string as an argument.
   c) concat two strings.
   d) convert case of string
   e) find one string from another.
6. Write a PHP program to find out maximum and minimum number.
7. Create an application to create a cookie, access a cookie and destroy the cookie.
8. Create an application that keeps track of how many times a visitor has loaded the page.
9. Set a session after user's login; maintain the user's data with session. Destroy the session and its data after a period of time.
10. Build an authentication application and restricts the unauthorized user from loading the page. And redirect the page with appropriate message
11. Write PHP code to store image in a database table.
12. Write PHP code to implement Querystring (passing variables using URL) concept.
13. Write PHP code to develop E-mail registration form and store all the submitted data in database table.
14. Write a program to read customer information like Cust_no, Cust_name, Item_purchased and Mob_no from Customer table and display all these information in table format on output screen.
15. Write a program to develop student registrationform and display all the submitted data on another page.
16. Write a program to read Employee informations like Emp_no, Emp_name, Designation and Salary from EMP table and display all these informations in table format on output screen. Provide option for editing and deleting a particular record from database.
17. Write PHP code to upload File/ image.
18. Write PHP code to download Data form Database into Excel, Word and PDF.
19. Develop an application which stores Railway's info with following fields Trainno, code,name, Type,Starting city, Destination city, Flag (W for weekly, D for Daily) Provide the following facilities like:
    a. Search by Starting city
    b. Search by Train Type
    c. List of train by Flag.
20. Write a program to calculate total weekly pay. If the user enters the number of hours worked and selects the hourly rate of pay from a list box. If overtime has been done, the number of hours is also entered. Over time hours are paid at double rate. A check box displays overtime. Calculate total amount to be paid.
21. Develop an application to add the movie name currently running with following operations:
    a. To see all the favorite movie
    b. To view top 5 and 10 movies
22. Create an application which displays the info about a particular institute which enables the user to see the faculty list according to department.
23. Write a PHP program to calculate interest for loan using user defined class 'loan calculator'.
24. Write a program for online merchants with following operations:

a. Customer login for further transactions
b. Validates the customer's information
c. System should protect customer's information

25. Develop an application for a shopping cart with following operations:
    a. Manage and display the catalog
    b. Add, Update and delete the products
    c. Process the shipping info
    d. Stores the order info
    e. Display the summary
26. Display the most popular item to your customer which is purchased the most? If the item is in top 5 display the description to the customer.
27. Create a database application for social gathering containing
    a. Information about the location (eg: club house, Party venue)
    b. Facilities available in the venue
    c. Booking for the specific events
    d. Display the booking details for current month and also generate the report for the bill to be paid
    e. for a particular booking
28. Create a Webpage which will create CD album (read data from JSON file) and convert data into HTML table.
29. Create a Webpage which will read data from JSON file display data into HTML table (use AJAX).
30. Create a Registration form with validation using AJAX.
31. Read data from XML file and Display CD catalog using AJAX. Provide navigation to move backward and forward.
32. Create a form validation program using jquery
33. Write a program to creating image slider using javascript.
34. Write a program using Ajax and JavaScript to get content of another file.
35. JQuery Ajax method to get content of another file 02
36. Write a program to swapping two images using javascript

2) Desirable: List of Practical's: AJAX & jQUERY

1) Create a polling application (Survey) using AJAX and PHP.
2) Create a Suggestion application using AJAX and PHP

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Basic Computer Concepts
Subject Code: 619406

With effective from academic year 2020-21

1. **Learning Objectives:**
   - **1)** To understand the fundamental concepts of Computers
   - **2)** To familiarize with components of computer systems
   - **3)** To understand working of computer and its application
   - **4)** To familiarize with modern Operating system concepts.

2. **Prerequisites:** Basic understanding of computer systems

3. **Course Contents:**

Unit	Course Content	Weightage Percentage
Unit I	**Introduction to Digital Logic:** Boolean algebra, Gates (AND,OR,NOT, NAND, NOR and XOR)	**3 (10%)**
Unit II	**Basics of Computer Architecture:** Organization and Architecture, Structure and Function, Evolution of the Intel x86 Architecture, Computer components, Computer Function, Interconnection Structures, Bus Interconnection, Point-to-Point Interconnect, USB, PCI Express	**5 (20%)**
Unit III	**Central Processing Unit :** Processor Organization, Register Organization, Instruction Cycle, CPU Bus Structure **Input and Output:** External devices, I/O Modules, Programmed I/O, Interrupt driven I/O, DMA	**7 (25%)**
Unit IV	**Introduction to Operating System and Systems Software's** Operating System Overview, Introduction to Scheduling Functions of Compilers / Interpreters, Linkers, Loaders, Cross Compilers (*)	**7 (25%)**
Unit V	**Computer Memory System Overview** Characteristics of Memory Systems, The Memory Hierarchy **Cache Memory:** Cache Memory Principles, Elements of Cache Design (Cache Addresses, Cache size, Mapping function, replacement algorithms, write policy, line size, number of caches) **Internal Memory:** Semiconductor Main Memory, Error correction, DDR DRAM, Flash memory, Newer non-volatile solid-state memory Technologies **External Memory:** Magnetic Disk, RAID, Solid State Drives, Optical Memory, Magnetic Tape	**6 (20%)**
	TOTAL	28 (100%)

Desirable:

- **Overview of Special-purpose Operating System:** Real Time Operating System (RTOS), Embedded Systems (Characteristics of Embedded Systems, TinyOS), Cloud Operating Systems, I oT Operating Systems

4. **Text Book:**
   - William Stallings, "Computer Organization and Architecture", 10th Edition, Pearson

5. Reference Books:

1. Morris Mano,Digital Logic and Computer Design, PHI
2. Andrew S. Tanenbaum, Todd Austin, Structured Computer Organization, 6th Edition, Pearson
3. Anita Goel,Computer Fundamentals, Pearson Education
4. William Stallings, "Operating Systems: Internals and Design Principles", 9th Edition, Pearson

6. Chapter wise Coverage from the Text Book:

Unit No.	Text Books	Topics/Subtopics
1	Book-I	Chapter 11 (11.1 to 11.2)
2	Book-I	Chapter 1 (1.1,1.2, 1.4), Chapter 3
3	Book-I	Chapter 14 (14.1 to 14.3), Chapter 7 (7.1 to 7.5)
4	Book I	Chapter 8 (8.1 to 8.3)
5	Book I	Chapter 4 (4.1 to 4.3), Chapter 5, 6
		Functions of Compilers / Interpreters, Linkers, Loaders, Cross Compilers - web sources

7. Accomplishment of the student after completing the course:

After completion of the course, students will be able to

- Explain the underlying processes of various topics and operations performed by them. For example,
  - When the system is started (booted), what happens;
  - Purpose of # include <stdio.h> in C and other libraries;
  - when the run command is given, what happens; etc
  - How arrays are internally stored?
- Utilize the internal architecture to understand the possibilities of optimizing the code and software

8. ACTIVE LEARNING ASSIGNMENTS:

- Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding of various
  - Operating Systems like Windows, LINUX, Ubuntu, Embedded Systems, Cloud OS, IoT OS, Mobile OS etc.
  - Topics from Book: Ron White, "How Computers Work", TechMedia Publications
- Compute the addition of array elements and calculate average using shared memory. Average will be calculated using barrier. Array is declared as shared resource.

- **Producer Consumer problem:**
  - Implement producer, consumer problem using bounde d buffer where array is shared. When producer will put the element in the array, consumer will wait until producer will release the array and same operation is applicable for consumer. Implement mutual conclusion using semaphore. When array

is full, producer will wait until consumer will consume at least one element. When buffer is e mpty, consumer will wait until producer will produce at least one element.

- Implement producer, consumer problem using un bounded buffer where array is shared. When producer will put the element in the array, consumer will wait until producer will release t he array and same operation is applicable for consumer. Implement mutual conclusion using semaphore. When buffer is empty, consumer will wait until producer will produce at least one element.

- **Reader Writer problem:**
  - More than one Reader can re ad from a file, but, when one writer will write, no other writer and readers can operate on the file. When one writer will write, whole file is locked by that writer. Implement these features using mutual exclusion.

- There is an array of elements of int. Generate histogram for calculating the occurrence of number of values within given range. Implement mutual exclusion and barrier in this application.

- There is an array of elements of int. Generate histogram for calculating the occurrence of number of values within given range. Implement mutual exclusion and barrier in this application.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications (2 years), 1st Semester
Bridge Course (1 week)

With effective from academic year 2020-21

Duration of Bridge Course: 1 Week (30 Hours)

1. Learning Objectives:

- The Bridge Course is aimed at providing the pre-requisites to newly admitted students so as to equip them to grasp the MCA subjects with ease. The pre-requisites are provided in the following three areas:
  - Computer Fundamentals
  - Basics of Mathematics
  - Website Design using HTML and CSS

2. Course Content

a. Part I : Computer Fundamentals (Total Hours : 10)

Sr#	***Course Content***
1	Introduction, Quick Overview of Computer Hardware, Arithmetic & Logical Operations Done by CPU, Importance of Programming: Use Basic Operations of CPU to Do Complex Tasks, Boot sequence
2	Basics of Flow Charting & Pseudo Code
3	Introduction to Number System
4	Boolean Algebra and Logic Gates
5	Introduction to Intel 8086 Architecture

b. Part II : Basics of Mathematics (Total Hours : 10)

Sr#	**Course Content**
1	**Set Theory** Daily life examples of sets, such as Tea Sets, Sofa Sets, etc. Introduce other examples of sets.
2	**Sequence** Daily life examples of sequence such as “Days of Week”, “Month Names”, etc. Introduce other examples of sequence. Explain the difference between Sets and Sequences.
3	**Matrices** Give example of Multiplication Tables of 2 to 10. 2 3 4 5 6 7 8 9 10 4 6 8 10 12 14 16 18 20 6 9 12 15 18 21 24 27 30 .. .. 18 27 36 45 54 63 72 81 90 20 30 40 50 60 70 80 90 100 …….. **Introduce the concept of matrices and give a few more examples.**
4	**Logic** Give day-to-day examples of simple statements. Through such examples, introduce the concept of symbolic representation of statements. Introduce the concept of Truth value of the statement.

5	**Integers** Formally introduce Natural Numbers, Integers. Describe factoring of a positive integer. Define Prime numbers and Composite numbers, and methods to find Prime numbers. Given 2 numbers, find their divisors (factors) and thereafter find common divisors. Take 2 or 3 symbols (a, b) or (a, b, c) or else digits such as (1, 2) or (1, 2, 3) and introduce the concept of Permutations and combinations.
6	**Relations** Using examples of family relations, introduce the concept of relations. Use other examples as well
7	**Functions** Draw graph of y = x + 5 and similar simple functions. Through such examples, introduce the concept of functions
8	**Graphs** Relations are set of ordered pairs. Draw these ordered pairs as edges and introduce the concept of graph
9	**Trees** Through Family Tree, Hierarchical Positions in a college (Principal → HoD → Faculty) introduce the concept of Tree. Take examples of hierarchical structure of vehicles (Vehicle → {4-Wheelers, 3-Wheelers, 2-Wheelers}, etc).Through such examples, introduce the concept of Class, Sub-class, Super-class, Objects, etc as well in addition to the concept of Trees

c. Part III : Website Design using HTML and CSS**(Total Hours : 10)**

Sr#	Course Content
1	**HTML5** HTML Basics: Introduction, Editing HTML5, First HTML5 Example, W3C HTML validation service, Headings, Linking, ImagesSpecial Characters and Horizontal Rules, Lists, Tables, Forms, Internal Linking, meta Elements HTML5: New HTML5 input Types, input and datalist Elements and auto complete Attributes, Page Structure Elements
2	**Cascading Style Sheets (CSS)** **CSS Basics**: Introduction, inline Styles, Embedded Style Sheets, Conflicting sheets, Linking External Style Sheets, Positioning Elements, Backgrounds, Element Dimensions, Box Model and Text flow, Media Types and Media Queries, Drop-Down Menus, Optional: User Style Sheets

Books:

i. Paul Deital Harvey Deitel, Abbey Deitel, Internet & World Wide Web: How to Program, 5th Edition, Pearson

Expected outcome:

- Students to develop static website for his/her BIO-Data using learned concepts for HTML5 and CSS.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Data Structures
Subject Code: 629401

With effective from academic year 2020-21

1. Learning Objectives:

- To develop proficiency in the specification, representation, and implementation of Data Types and Data Structures.
- To introduce the concepts of algorithmic paradigms and basic data structures and their applications.
- To analyze various algorithms for space and time complexity.
- To implement and compare various searching and sorting techniques.
- To apply appropriate data structures to solve different problems.
- To develop a base for advanced computer science study.

2. Prerequisites:

Proficiency in a programming language

3. Course Contents:

Unit No.	Course Content	Weightage Percentage
I	**Introduction to Data Structure and Algorithm Analysis:** Data Structure Definition and classification, Algorithm Analysis, Storage Representation of Strings, Text Handling and KWIC Indexing.	**10%**
II	**Linear Data Structures:** Arrays, Storage Structure for Arrays, Stack: List Implementation, Applications of Stacks: Function Call, Recursion, Balancing Symbols Queue: List Implementation, Circular Queue, Priority Queue, double ended queue. Linked List: Cursor Implementation, Multi List Applications of Linked List : Addition and Multiplication of Polynomial in one and two variables	**25%**
III	**Nonlinear Data Structures:** Tree - Basic Tree Concepts, Operations on Binary Trees, Storage Representation & Manipulation of Binary Trees, Conversion of General Tree to Binary Trees, Sequential & Other Representation of Trees, Application of Trees – The Manipulation of Arithmetic Expression, Multi-linked Structures - Sparse Matrices.	**20%**
IV	**Graphs and Their Representation**: Matrix Representation of Graphs, List Structures, Other Representation of Graphs, Breadth First Search and Depth First Search.	**20%**
V	**Sorting and Searching Techniques:** Sorting – Notation and Concepts, Selection Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort, Searching - Sequential Searching, Binary Searching, Search Trees – Height Balanced, 2-3 Trees, Weight Balanced Tree, Tree Structures, Hash Table Search Methods, Introduction, Hashing Functions.	**25%**

4. Text Books:

1. Jean-Paul Tremblay, Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill, 2nd Edition, (2007)
2. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, Pearson, Second Edition

1. Reference Books:

- Ashok N. Kamthane, “Introduction to Data Structures in C”, Pearson Education (2004).
- Cormen, Leiserson, Rivest, Stein,"Introduction to Algorithm", PHI. 2nd Edition(2003).
- Parag H Dave, Himanshu B Dave, “Design and Analysis of Algorithms”, Pearson (2014)
- Samir Kumar Bandyopadhyay, Kashi Nath Dey, “Data Structures Using C”, Pearson Education, Year: 2004.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education (2002).
- Horowitz, Sahni, Anderson-Freed, “Fundamentals of Data Structures in C”, University Press (2nd edition-2007)
- G. A.V.PAI, "Data Structures and Algorithms, Concepts, Techniques and Applications”, TMH, 1st Edition (2008).

2. Chapter Wise Coverage from Text Book:

Unit No.	Text Books	Topics/Subtopics
I	Book-1	0-3.0 to 0-3.5, 2.4, 2.5.3
II	Book-1 Book-2	3.2, 3.5,3.6 to 3.8,4.3.1 3.3.3,3.2.7,3.2.8
III	Book-1	5.1.1 to 5.1.5, 5.2.1, 5.3.1
IV	Book-1	5.4.1 to 5.4.5
V	Book-1	6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.1.6, 6.2.1, 6.2.2, 6.2.3,6.2.3.1, 6.2.3.2, 6.2.3.3, 6.2.3.4, 6.2.4, 6.2.4.1, 6.2.4.2

3. Accomplishments

- Apply sorting and searching algorithms to the small and large data sets.
- Ability to design and implement abstract data types such as linked list, stack, queue, graphs and trees using static or dynamic implementations.
- Analyze the complexity of different algorithms.

Practical List

Use C programming language to perform followings Lab work:

1.	Create a Structure with following Data Members: 1. Integer Array 2. Size of the Array Sort the Array using various Sorting algorithms such as (i) Selection Sort (ii) Bubble Sort (iii) Two-way Merge Sort (iv) Quick Sort (v) Heap Sort. And store the sorted Array in a text file.
2.	Create a Structure with following Data Members: 1. Integer Array 2. Size of the Array Search an element in Array using Linear (Sequential) Search and Binary Search, and Display result in file. For Sequential Search, assume that input array is Unordered and for Binary Search assume that input array is Ordered and develop programs accordingly.
3.	Create a "Stack" data structure with following Data members: 1. Integer Array 2. Stack Pointer (Top of Stack: Is it same as the Size of the Array) Perform the following operations on stack using user-defined functions: 1. Push 2. Pop 3. Isempty 4. Isfull 5. Peep Create a file which stores all values of Array through Stack. Has it reversed the order of the elements of the Array? Why?
4.	Create a "Linked List" structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on stack using user-defined functions: 1. Insert a value X at the first place 2. Insert a value X at the end of the list 3. Insert a value X at the place so that it preserves the ordering of the terms in the increasing order. i. Delete an element whose address is given by X i. Copy a linked linear list Create a file which stores all values of list.
5.	Write a program to convert an infix arithmetic expression (parenthesize / unparenthesized) into postfix notation.
6.	Write a program to evaluate a postfix expression.
7.	Create a structure with the following Data members: 1. Integer Array 2. Size of the Array Search an element in a given list using Binary Search by recursion. And Display result in a file.

8.	Create a "Queue" structure with following Data members: 1. Integer Array 2. Size of the Array Perform the following operations on Simple queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of Array.
9.	Create a "Queue" user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Simple queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of list.
10.	Create a "Circular Queue" structure with following Data members: 1. Integer Array 2. Size of the Array Perform the following operations on Circular queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of Array.
11.	Create a "Circular Queue" user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Circular queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of list.
12.	Create a user-defined "Linked List" structure with the following data members: 1. A Co-efficient 2. An Exponent 3. A link to the next node Perform the following operations on Singly list using user-defined functions: 1. Create 2. Display 3. Addition 4. Multiplication Create a file which stores all values of list.

13.	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on list using user-defined functions: 1. Create a list 2. Traverse the whole list 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.
14.	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Circular list using user-defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.

15.	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node 3. A link to the previous node Perform the following operations on the doubly-linked list using user-defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.
16.	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node 3. A link to the previous node Perform the following operations on doubly-linked Circular list using user defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.

17.	Write a program to represent an undirected graph using the adjacency matrix to implement the graph and perform following operations, with menu driven options for following tasks: 1. Create graph 2. Insert an edge 3. Print Adjacency Matrix 4. List all vertices that are adjacent to a specified vertex. 5. Print out vertices using depth first search 6. Print out vertices using breadth first search 7. Exit program
18.	Create a user-defined structure with the following data members: 1. A Data 2. A link to the Left child 3. A link to the Right child Perform the following operations on Binary Search Tree using recursion: 1. Create 2. Traverse (Inorder, Preorder, Postorder) 3. Insert 4. Delete 5. Search 6. Create a file which stores all values of traversal.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Mobile Computing using Android
Subject Code: 629402

With effective from academic year 2020-21

1. Learning Objectives:

- To be able to understand the process of developing software for the mobile
- To be able to create mobile applications on the Android Platform
- To be able to create mobile applications involving data storage in SQLite database.

2. Prerequisites: Knowledge of the Core Java Programming, database concepts

3. Contents:

Unit No.	Course Content	Weightage Percentage
I	**Basics of ANDROID** Introduction to ANDROID: ANDROID SDK Features, Introduction to Development Features Developing for ANDROID, developing for mobile and embedded devices, ANDROID development tools Creating Applications using ANDROID Basics of an ANDROID application, introduction to manifest, externalizing resources, application lifecycle, ANDROID activities	**10%**
II	**User Interface in Android** Building user interfaces Introduction to layouts, introduction to fragments, creating new views, introduction to adapters Intents and broadcast receivers Introduction to intents, creating intents and broadcast receivers Using Internet resources Downloading and parsing internet resources, using the download manager, using internet services	**25%**
III	**Working with Data in Android** Files, saving state and preferences Creating, saving and retrieving shares preferences, including static files as resources, working with the file system **Database and content providers** Introducing ANDROID databases, content values and cursors, working with SQLite databases, creating content providers, using content providers, native ANDROID content providers	**25%**

	Working in background Introducing services, using background threads, using alarms	
IV	**Enhancing user experience** Introduction and addition of action bar, Creating and using menus and Action bars action menus, introducing dialogs, let us make toast, Introducing notifications, **Advanced User experience:** Designing for every Screen size and density, Introducing Android Text-to speech, Using speech recognition, Controlling Device vibration, Working with Animations, Enhancing your views, Advanced Drawable resources, Copy, paste and the clipboard.	**15%**
V	**Advanced Android** **Audio, video and using the camera:** Playing audio and video, manipulating raw audio, using camera to take pictures, recording video, adding media to media store **Telephony and SMS:** Hardware support for telephony, using telephony, introducing SMS and MMS	**15%**
VI	**Publishing your App:** Monetizing, promoting and distributing the applications, Signing and publishing applications, distributing applications, introduction to monetizing applications	**10%**
VII (*)	**Web Services (Only for implementation in lab)** **Introduction, Web services Types, Consuming Web Services, Performance** **Case Study:** Simple Android Web service, How to call Java web services in Android, Android JSON or XML web services, Authenticate users using web service	-

(*) Unit 7 is only for Practical implementation in App development.

Desirable Topics:

1) Navigation drawer. recycler view and material design: https://developer.android.com
2) Performance Improvement of App, Location services, Google maps API, Google Places API
3) Android 9 Overview (Kotlin)

4. Text Book:

1. Reto Meier, Professional ANDROID 4 Application Development, WROX Latest Edition
2. Nolan Godfrey, Onur Cinar, David Truxall, Android Best Practices, A press

5. Reference Books:

1. Lauren Darcey and Shane Conder, "Android Wireless Application Development", Pearson Education, 2nd ed. (2011)
2. Mark L Murphy, "Beginning Android", Wiley India Pvt Ltd (2009)
3. Sayed Y Hashimi and Satya Komatineni, "Pro Android", Wiley India Pvt Ltd (2009)

Web references

- https://developer.android.com/index.html
- Android Developer Fundamental Course by Google and Advance Android Developer Course by Google (https://developer.android.com/courses)

6. Chapter wise Coverage from Text Book:

Unit No	Book#	Chapters
1	1	Chapter 1,2,3
2	1	Chapter 4,5,6
3	1	Chapter 7,8,9
4	1	Chapter 10,11
5	1	Chapter 15, 17
6	1	Chapter 19
7	2	Chapter 8

Tools/Technologies to used be:

1 ANDROID Studio [Latest Version]
2. ANDROID Version [Jelly Bean and later]

7. Accomplishments of the student after completing the course:

Student will visualize the real world mobile application scenario and enables them for development and implementation of mobile applications

8. Suggestions for Lab Sessions: (Group Activity)

General Guidelines

1. Group: 2-3 Person.
2. The project should be free from plagiarism of any kind.
3. It is mandatory that the project should be developed using Android Studio 2 later version on Linux or Windows Platform.
4. This may not be a live project

B. Expected Outcome

- The objective of the Mobile Application development using Android is to make students aware about the industry based process and workings. As a result, Application must meet with the industry standards.
- There will not be any compulsion to prepare a project report for the students but an application should be demonstrated, so that evaluator may get the detail about the Project developed and can evaluate the students as per the evaluation criteria.
- Each group to develop 2 applications with minimum features describe as below:
  - **Sample Definition for Simple Application:**
    - **Learning objective:** Idea is to make students learn the basic of android by imitating the apps that are easy to build and uses some logic

    1. Number Guessing Game [User will guess a number from 1 to 100 and will be compared with random number generated]
    2. This can be extended to store the score in file [+1 to score if number matches]
    3. Dice Rolling Game [Random generating numbers and displaying image like dice]
    4. Simple Calculator [This can be extended to save the whole calculation sequence so that it can be rechecked]
    5. To-Do List App [Storing and reading file]
    6. Timer [This can be extended to store last 10 timer details]
    7. Daily Diary [To store the date and description to write daily experience]
    8. Daily Motivational Thought / Article [Fetching the article for Wikipedia, thought from some site or file]

    Advanced Application 2: Web service Based

    - Patient Appointment App,
    - Movie Ticket Booking
    - Cab Booking Android Application
    - ST Bus Ticket Booking
    - Vehicle tracking system
    - Attendance System
  - More project definition can be found at https://nevonprojects.com/project-ideas/android-project-ideas/.

Evaluation

Sr. No.	Particulars	Weightage
1	**Application – 1 Simple**	**20%**
2	**Application – 2 – Advanced**	**30%**
3	**Code changes for Application – 2**	**30%**
4	**VIVA**	**20%**

Preferably application may be uploaded on app store

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Programming in Python
Subject Code: 629403

With effective from academic year 2020-21

1. Learning Objectives:

- To develop proficiency in creating based applications using the Python Programming Language.
- To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.
- To be able to draw various kinds of data visualization techniques using PyLab, matplotlib and Pandas
- To be able to understand the creation DB API in Python

2. Prerequisites: Basic Concept of Programming Language

3. Contents:

Unit	Course Content	Weightage percentage
Unit I	**Introduction to Python** The basic elements of Python, Objects, expressions and numerical Types, Variables and assignments, IDLE Branching programs, Strings and Input, Iteration Structured Types, Mutability and Higher-order Functions: Tuples, Ranges, Lists and Mutability (Cloning and list comprehension), Strings, Tuples and Lists, Dictionaries	**15%**
Unit II	**Functions, Exception, Modules and Files** **Functions:** Difference between a Function and a Method, Defining a Function, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas (Using Lambdas with filter() Function, Using Lambdas with map() Function, Using Lambdas with reduce() Function), Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python, The Special Variable __name__ **Exceptions**: Errors in a Python Program (Compile-Time Errors, Runtime Errors, Logical Errors), Exceptions, Exception Handling, Types of Exceptions, The Except Block, the assert Statement, User-Defined Exceptions, Logging the Exceptions 20% **Files:** Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, Knowing Whether a	**25%**

	File Exists or Not, Working with Binary Files, The with Statement, Pickle in Python, The seek() and tell() Methods	
Unit III	**Classes and Object-oriented Programming** Abstract Data Types and classes, Inheritance, Encapsulation and Information hiding Mortgages and Extended Examples Case Study: Banking Application	**20%**
Unit IV	**Advanced Topics I: Data Science and Data Visualization using Python** **Data Science Using Python:** Data Frame (Creating Data Frame from an Excel Spreadsheet, Creating Data Frame from .csv Files, Creating Data Frame from a Python Dictionary, Creating Data from Python List of Tuples, Operations on Data Frames) **Data Visualization:** Bar Graph, Histogram, creating a Pie Chart, Creating Line Graph **Plotting:** Plotting using PyLab, Plotting mortgages and extended examples	**15%**
Unit V	**Advanced Topics II: Regular Expressions** **REs and Python:** Regular Expressions, Sequence Characters in Regular Expressions, Quantifiers in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expressions on Files, Retrieving Information from a HTML File Case Study: Screen Scrapping	**25%**
Unit VI (*)	**Python's Database Connectivity** Verifying the MySQL dB Interface Installation, Working with MySQL Database, Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Rows from a Table, Updating Rows in a Table, Creating Database Tables through Python	**-**

Note: (*) Only for Practical exam.

4. Text Book(s):

John V Guttag. "Introduction to Computation and Programming Using Python", 2nd Edition, Prentice Hall of India

1. R Nageswara Rao, Core Python Programming, 2nd Edition, Dreamtech Press

5. Reference Books:

1) Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson
2) Luke Sneeringer, Professional Python, WROX
3) Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python, Pearson

4) Doug Hellmann, The python 3 standard Library by example, Pearson Education
5) Alex Martelli, Python Cookbook, O'REILLY
6) Laura Cassell, Python Projects, WROX
7) Daniel Y Chen, Pandas for Everyone: Python Data Analysis, 1st Edition, Pearson Eduction

Web Resources:

1) Charles Severance, Python for informatics: www.pythonlearn.com
2) Swaroop C H. "A Byte of Python", http://www.swaroopch.com/notes/python
3) "Python Programming", http://en.wikibooks.org/wiki/Python_Programming
4) "The Python Tutorial", http://docs.python.org/release/3.0.1/tutorial/
5) "Learn Python the Hard way", http://learnpythonthehardway.org/
6) Dive into Python 3: http://www.diveintopython.net/

6. Unit wise coverage from Textbook(s):

Unit #	Book#	Chapter
I	1	Chapter : 2,5 (Except 5.4)
II	2	Chapter :9,16, 17(Pages 441 to 456)
III	1	Chapter : 8
IV	1	Chapter : 11
	2	Chapter : 25
V	2	Chapter : 18
VI	2	Chapter : 24 (Pages 663 to 681)

Accomplishment

At the end of the course, the student should be able to:

- Ability to create robust applications using the Python programming language
- Ability to create applications for solving computational problems using the Python Programming Language

Practical List

Tools: Python 3.x, IDLE

Part I: Core Python

A Basics

1	Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa.
2	Write a program in python to swap two variables without using temporary variable.

3	Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
4	Write a program to make a simple calculator (using functions).
5	Write a program in python to find out maximum and minimum number out of three user entered number.
6	Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.
7	Write a Python program to check if the number provided by the user is an Armstrong number.
8	Write a Python program to check if the number provided by the user is a palindrome or not.
9	Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Count Length of string (do not use Len ()) c) Reverse string d) Find and replace operation e) check whether string entered is a palindrome or not
10	Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following: **** ********* *******
11	Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function)
12	Write a program in python to implement Factorial series up to user entered number. (Use recursive Function)
13	Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.

B. **Advanced**

14	Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip for given Employee Number b) Print Employee List for Given Department Number
15	Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations. a. Reserve a ticket for a passenger. b. List information all reservations done for today's trains.
16	Write a Python program to implement module.
17	Write a program which will implement decorators for functions and methods in python.
18	Write a program to read CSV file and generate output using HTML table.
19	Write a program to process CSV file using CSV module.
20	Desirable: Write a program to process JSON and XML data.

21	Create Web Database Application "Address Book" with options to a) add/ insert a record b) modify a record c) display a record d) delete a record
22	Create Web Database Application "Event Registration" with options to a) Event Registration b) Cancel Registration c) display a record

Part II: Advanced Topic: Data Analysis

1	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Data Analysis using groupby()

Part III: Advanced Topic: Data cleaning

1	Handling dirty data / missing data

Part IV: Advanced Topic: Python for Data Visualization
Library: pylab, matplotlib, seaborn

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
2	Using a data file, draw a. Bar Graph b. Histogram c. Pie Chart d. Line Chart
3 (*)	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Handling missing data

(*): Topics from Books 2 (Pages 694 to 705)

Part V: Advanced Python Programming: Regular Expressions

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
	a) Recognize following strings bit, but, bat, hit, hat or hut
	b) Match any pair of words separated by a single space, that is, first and last names.
	c) Match any word and single letter separated by a comma and single space, as in last name, first initial.
	d) Match simple Web domain names that begin with www and end with a ".com" suffix; for example, www.yahoo.com. Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example: www.foothill.edu).
	e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 De la Cruz Boulevard.
2	Create utility script to process telephone numbers such that
	a. Area codes (the first set of three-digits and the accompanying hyphen) are optional, that is, your regex should match both 800-555-1212 as well as just 555-1212.
	b. Either parenthesized or hyphenated area codes are supported, not to mention optional; make your regex match 800-555-1212, 555-1212, and also (800) 555-1212.
3	Chapter End Practical List of Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Computer Networking
Subject Code: 629404

With effective from academic year 2020-21

1. Learning Objectives:

- To equip the students with basics concept of Computer Networks.
- To familiarize the students with the standard models for the layered approach OSI and TCP/IP to communication between machines in a network and the protocols of the various layers.
- To gain basic insight of programming for network solutions.

2. Prerequisites:

Basic Concept of Data Structures, Operating systems, Programming Language and Knowledge of Linux OS

3. Contents:

Unit	Course Content	Weightage percentage
Unit I	**Introduction:** Personal Area Network, Local Area Networks, Metropolitan Area Networks, Wide Area Networks, Internetworks, Network software, protocol hierarchies, Design issues for the layers, connection oriented vs. Connectionless service, service primitives, relationship of services in protocols, Reference Models, Open System Interconnection (OSI), TCP/IP Reference models, Example Networks: ARPANET, NSFNET, Architecture of Internet.	**15%**
Unit II	**Physical Layer** Guided Transmission Media, Magnetic Media, Twisted Pairs, Coaxial Cable, Power Lines, Fiber Optics, Wireless Transmission, Electromagnetic Spectrum, Radio Transmission, Microwave Transmission, Infrared Transmission, Light Transmission, Digital Modulation and Multiplexing, Public Switched Telephone Networks.	**15%**
Unit III	**Data Link Layer** Design issues, Error detection and correction. Elementary data link protocols: Utopian simplex protocol, a simplex stop and wait protocol for an error-free channel, a simplex stop and wait protocol for noisy channel. Sliding Window protocols, Example data link protocols.	**15%**
Unit IV	**Medium Access Control Sublayer** The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, IEEE 802.11 Architecture and Protocol	**15%**

	Stack, Physical layer, sublayer, Frame Structure, Services, Data link layer switching.	
Unit V	**Network Layer** Design issues, Routing algorithms: Optimality principle, shortest path routing, Flooding, distance vector routing, Link State routing, Congestion Control Algorithms, The Network layer in the Internet.	**15%**
Unit VI	**Transport Layer:** Transport Services, Elements of Transport protocols, Connection establishment, connection release, Error control, flow control, congestion control, UDP and TCP protocols.	**15%**
Unit VII	**Application Layer:** Domain name system, Electronic Mail; the World Wide Web, HTTP	**10%**

4. Text Book(s):

Andrew S Tanenbaum, David. J. Wetherall, "Computer Networks", Pearson Education, 5th Edition,

5. Reference Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw-Hill, Fourth Edition
2. Kurose and Ross, Computer Networking- A Top-Down approach, Pearson, 5th edition
3. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann Publishers, Fifth Edition, 2011.
4. Fred Halsall, Computer Networking and the Internet, Addison Wesley, (5th edition)
5. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", Mc Graw Hill Publisher, 2011
6. Bhushan H Trivedi, "Computer Networks", Oxford University Press

6. Unit wise coverage from Textbook(s):

Unit 1	**Topics**
I	Chapter 1: 2,3,4.1 to 4.4,5.1
II	Chapter 2: 2,3,5,6
III	Chapter 3: 1,2,3,4,5
IV	Chapter 4: 1,2,3.1 to 3.7,4,5,6
V	Chapter 5: 1,2.1 to 2.5,3,6.1 to 6.4
VI	Chapter 6: 1.1,1.2, 2.1 to 2.4, 3.1 to 3.2, 4.1, 4.2,5.1 to 5.9
VII	Chapter 7: 1,2,3.1, 3.4

Students are not required to reproduce the entire algorithms/protocol code in the theory exam for any protocols and routing algorithms. Concepts based on these algorithms/ protocols should be asked in theory exam

7. Outcomes

At the end of the course, the student should be able to:

1. Understand concepts of networking and gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
2. Identify the components required to build different types of networks
3. Obtain the skills of subnetting and routing mechanisms.
4. Have a working knowledge of datagram
5. Trace the flow of information from one node to another node in the network

Practical List

Part I: Network Basics

A. Hardware

1. Install & Test Network Interface Card.
2. Prepare and Test Straight UTP Cable.
3. Prepare and Test Cross UTP Cable.
4. Develop a small Network. (Hands on Training.)

B. Software

5. Install Windows 2003/Windows 2008 Network operating System
6. Install & Configure File Server.
7. Install & Configure Print Server
8. Install & Configure Mail Server
9. Install & Configure Proxy Server
10. Install & Configure Web Server
11. Install & Test Router, Repeater and Bridge.
12. Install a small wireless network

Part II: Network Analysis using Wireshark

1. Download Wireshark from its official webpage, it is a network packet analyser https://www.wireshark.org/
2. Install Wireshark under Windows/Linux/MAC/Solaris platform, Windows installer names contain the platform and version. Install WinPcap

Lab1:

Objective: To make the students aware about the IT/Network infrastructure of their parent institute.

Introduction to the lab infrastructure, the cabling/cable type and specifications/ switch/ topology/ router/network infrastructure, Internet Connectivity, Wi-Fi Connectivity of your department (labs and lecture halls) and how it is connected with the overall institute level IT infrastructure, How the institute obtains Internet Connectivity from its ISP, The Bandwidth and other specifications, Visit to the Computer Centre of the Institute, Introduction to various Enterprise Servers/Servers of Licensed Software, Server hardware configuration, Server Management Tools, Enterprise Level Firewall, etc. So, lab 1 is intended as an in-

campus IT infrastructure industrial visit for students. Students are supposed to prepare a report of this visit. Students are also supposed to note down the model/make of various network interface devices (NIC card, switch, router) used in their lab/department/institute, download their technical specifications from their respective web-site and attach those specifications in the report. The significance and interpretation of these specifications may be discussed by the faculty member during the corresponding lab/lecture hours pertaining to that particular device.

Lab 2:
Objective: To make the students aware about and learn the detailed use of the following OS level TCP/IP diagnostic and troubleshooting commands: ping, ns lookup

Lab 3:
Objective: To make the students aware about and learn the detailed use of the following OS level TCP/IP diagnostic and troubleshooting commands: ipconfig, arp, netstat, tracert, telnet

Lab 4a:
Objective: Implement a simple TCP socket based client server program in Python in which the
client connects to the server. The server displays the ip address and port number of client and
sends an acknowledgement message back to client. The client displays the received acknowledgement message on screen.

Lab 4b:
Objective: Implement a simple UDP socket based client server program in Python in which the
client connects to the server. The server displays the ip address and port number of client and
sends an acknowledgement message back to client. The client displays the received acknowledgement message on screen.

Lab 5:
Objective: Wireshark Installation, understanding the Wireshark Environment / Menu System,
to start and stop live capture of traffic from given wired Ethernet network interface, capturing
options, store captured data in different supported file formats, to open already stored captured
data file.

Lab 6:
Objective: Learn and use view level filters and capture level filters in Wireshark for different
traffic types like Ethernet, ARP, IP, TCP, UDP, DNS, HTTP, etc. For each captured category,

observe how different headers are encapsulated within each other. E.g. TCP encapsulated within IP, HTTP encapsulated within TCP, etc.

Lab 7:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: Ethernet and ARP

Lab 8:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: IP and ICMP

Lab 9:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: TCP and UDP

Lab 10:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: HTTP and DNS

Lab 11:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: FTP, SMTP and Telnet

Lab 12:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header
and payload: Ethernet and ARP

Lab 13:
Objective: Capture Wi-Fi and Bluetooth Traffic and Interpret/ Analyze the corresponding header and payload using Wireless Traffic Sniffing tools like Wireshark-USB/AirCrackng/ Kismet, etc.

Lab 14:
Objective: Analyze Email Traffic: Normal POP Communications, POP Problems, Dissect the
POP Packet Structure, Filter on POP Traffic, Normal SMTP Communications, SMTP Problems, Dissect the SMTP Packet Structure, Filter on SMTP Traffic

Lab 15:

Objective: Analyze IEEE 802.11 (WLAN): Wireless LANs (WLANs) Traffic, Signal Strength
and Interference, Capture WLAN Traffic, 802.11 Traffic Basics like Data Frame, Normal 802.11 Communications

The Practical Examination Exercises would be based on Lab Exercises based on above list EXCEPT Lab # 1, 4a, 4b, 13 and 15. For Lab # 1, 4a, 4b, 13 and 15, documentation/report/file
should be maintained.

References:

1) https://technet.microsoft.com/en-us/library/bb727023.aspx
2) https://docs.python.org/2/howto/sockets.html
3) https://www.aircrack-ng.org/
4) https://www.kismetwireless.net/

Note: Some of the practical's form the above practical list may have seemingly similar definitions. For better learning and good practice, it is advised that students do maximum number of practical's. In the practical examination, the definition asked need not have the same
wordings as given in the practical list. However, the definitions asked in the exams will be similar to the ones given in the practical list.

Part III: Exploring using Python

1. Implement a Python Program to print host name and IP address of local host.

2. Implement a Python Program to print host name and IP address of remote host where IP address of remote host is available.

3. Implement a Python Program to print host name and IP address of remote host where hostname of remote host is available.

4. Implement a TCP port scanner program in python for local host.
(Note: Do not try this program for a remote host, especially outside your domain. It could cause legal problems)

5. Implement a UDP port scanner program in python for local host.
(Note: Do not try this program for a remote host, especially outside your domain. It could Cause Legal problems)

6. Implement a TCP based client server program in python using TCP sockets where Server displays the following: a) Host Name, IP address and Port Number on which it is hosted b) IP address and port number of a client requesting connection. Server sends the message "Thanks for Connecting!" back to client. Client displays this message on screen.

7. Implement a UDP based client server program in python using UDP sockets where Server displays the following:
   a) Host Name, IP address and Port Number on which it is hosted
   b) IP address and port number of a client sending some dummy message. Server displays the dummy message on screen. Server sends the message "Thanks for Message!" back to client. Client displays this message on screen.

8. Implement a TCP based echo client server program in python.

9. Implement a UDP based echo client server program in python.

10. Implement a TCP based daytime client server program in python.

11. Implement a UDP based daytime client server program in python.

12. Implement a TCP based client server text chat program in python.

13. Implement a UDP based client server text chat program in python.

14. Implement a TCP based echo client server program in python with a multi-threaded server.

15. Implement a TCP based daytime client server program in python with a multi-threaded server.

16. Implement a web client using urllib to:
   a) Display the html source of a given URL on screen
   b) Display the URL visited
   c) Display the header information transmitted in the http response sent by the contacted web-site/web-server.
   d) Display the http server status code

17. Implement an ftp client using ftplib which connects to an ftp server, takes login/password from user, displays directory list and permits upload and download of files.

18. Write a Python program that makes a connection to a web server and retrieve/display a document using the HTTP protocol.

19. Write a Python program that makes a connection to a web server and retrieve an image using the HTTP protocol.

20. Write a python program to implement a simple server-client program :

21. Write a python program to implement socket programming using multi-threading

Reference Material for Python Network Programming:

1. Learning Python Network Programming, Dr. M.O. Faruque Sarker & Sam Washington, Packt Publishing, 2015
2. Python Network Programming Cookbook, 2nd Edition, Pradeeban Kathiravelu, Dr. M.O. Faruque Sarker, Packt Publishing, 2017
3. Foundations of Python Network Programming, 3rd edition, Brandon Rhodes & John Goerzen, Apress Publications
4. Beginning Python: From Novice to Professional, 2nd edition, 2009 (Indian Reprint), Apress Publications
5. https://docs.python.org/3/howto/sockets.html
6. https://docs.python.org/3/library/socket.html

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Software Project-1
Subject Code: 629405

With effective from academic year 2020-21

1. Learning Objectives:

- To solve industrial (or society or research) problems.
- To plan, schedule, and monitor the software project
- Development, coding, and testing of a large project cohesively.
- Documentation of project

2. Prerequisites:

Software Engineering, Programming / Coding language, RDBMS

3. Guidelines for Project

- It is suggested to develop Web applications using emerging Web frameworks like LARAVEL, DJANGO, ASP.NET, XAMARIN, Spring or ecommerce etc.
- Mobile Application is not allowed for development.
- It is recommended that the team should be about 2-3 students.
- The project should be free from plagiarism of any kind.
- Internal guides (i.e. The regular faculty members) must be allocated to projects.
- Project plan along with the division of work amongst teammates would have been prepared and got approved within a maximum of 5 days of the start of the project.
- Coding standards should be followed meticulously. At the minimum, the code should be self-documented, modular, and should use the meaningful naming convention.
- It is advisable that object-oriented methodology is used with the reusability of classes and code, etc.
- The output reports must include MIS reports, if applicable.
- The documentation should include a chapter on "Learning during Project Work", i.e. "Experience of Journey during Project Duration".
- It is strongly recommended that Data structure/Database design is included in the report. At least portions of code (preferably full code) are mandatory. The student may be asked to write the code related to the project during the examination.
- If a student is compelled to follow certain instructions (by the external, i.e. organization's Guide) which he/she does not agree to, such a student must prepare a supplementary report to document his/her version and present it to the examiners if such a need arises.
- Internal guides (i.e. The regular faculty members) must devote the time allocated as per the timetable to guide the students for the project. The time allocation will be in accordance with the scheme for the 6th semester project as given.
- Internal guides should preferably visit external guide to track the project
- Project document should be printed on both sides of paper.

4. Accomplishments of the student after completing the course:

- Doing the project will enable the student to go through rich experience in developing large projects. Such an experience will include encountering various technical issues, finding sources to resolve the issues and finally finding the solution of all these issues satisfactorily.
- Thinking analytically, analysing, and synthesizing requirements and complicated information for getting a good comprehension of the solution methodology to be adopted.
- Ability to document and write well.
- Organizing the time effectively.

- Working with teammates and generating substantial output of the efforts.
- It will prepare the students for analysing and programming for industrial problem and large projects work in future

5. Documentation:

- The project has to be well-documented in the form of a Project Report (at least 40 pages comprising of the design, data dictionary, source code, screenshots, etc.).
- Format: Print out on both the side of page with single line spacing. Use Times New Roman of size 10 for normal text.

TABLE OF CONTENTS

1. Introduction
    1.1. Existing System
    1.2. Need for the New System
    1.3. Objective of the New System
    1.4. Problem Definition
    1.5. Core Components
    1.6. Project Profile
    1.7. Assumptions and Constraints
    1.8. Advantages and Limitations of the Proposed System
2. Requirement Determination & Analysis
    2.1. Requirement Determination
    2.2. Targeted Users
3. System Design
    3.1. Use Case Diagram
    3.2. Class Diagram
    3.3. Interaction Diagram
    3.4. Activity Diagram
    3.5. Data Dictionary
4. Development
    4.1. Coding Standards
    4.2. Screen Shots
6. Proposed Enhancements
7. Conclusion
8. Bibliography

6. Evaluation Parameter:

- Evaluation of the projects would be done considering the framework available at the Institute. The main parameter of assessment would be the ability of the students to code.
- Though the project and domain specific knowledge would be assessed for, the evaluation would predominantly depend on the students' ability to explain, modify or revise of code.
- Coding standards should have been implemented.
- Though the project would be evaluated for the entire team, the examiner should emphasize on the contribution of each team member in the project development

Sr. No	Particulars	Weightage
1	Project (definition, functionality and execution)	30%
2	Documentation	20%
3	Code Changes	25%
4	VIVA	25%

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Software Testing
Subject Code: 629406

With effective from academic year 2020-21

1. Learning Objectives:

- To understand the importance of Software Testing for good Quality Software
- To understand Software Quality Assurance (SQA) architecture and its importance with the details of each component.
- To Design, Develop and write Test Plan & Test Cases
- To automate testing with Selenium Web driver

2. **Prerequisites:** Software Engineering Basics, Basics of Java Programming

3. Contents:

Unit	Chapter Details	Weightage Percentage
Unit I	**Introduction to software Quality and Assurance** The software quality challenge, Software quality, Software quality factors Management and its role in software quality assurance	**10%**
Unit II	**Components of SQA** The components of the software quality assurance system – overview **Pre-project Software Quality Components** Contract review, Development and quality plans	**15%**
Unit III	**SQA Components in the Project Life Cycle and Strategies** Integrating quality activities in the project life cycle, Reviews, Software testing – strategies, Black Box Testing and White Box Testing	**25%**
Unit IV	**Software Testing –Implementation**: The Testing Process, Test Case Design, Automated Testing, Testing Life Cycle, Software Quality Implementation, Assuring the quality of software maintenance components, Assuring the quality of external Participants' contributions, CASE tools and their effect on software Quality	**25%**
Unit V	**Software Quality Infrastructure Components (Introduction only)** Procedures and work instructions, Staff training and certification, Corrective and preventive actions, Documentation control	**10%**
Unit VI	**Software Quality Metrics** Software Quality metrics, Cost of Quality	**15%**
Unit VII (*)	**Introduction to Selenium, Selenium WebDriver, Selenium WebDriver Architecture, Selenium Test Life Cycle**	•

(*) – Only for practical.

4. Text Book:

Daniel Galin, "Software Quality Assurance", Pearson Publication, 2009.

5. Reference Books:

1. KshirsagarNaik and PriyadarshiTripathy, Software Testing & Quality Assurance-Theory and Practice, Wiley Studentedition
2. William E. Perry, Effective Methods for Software Testing, WILLEY, 3rdEdition
3. Alan C. Gillies, "Software Quality: Theory and Management", International Thomson Computer Press,1997.
4. M G Limaye, Software Testing, Tata McGraw-Hill Education,2009

6. Chapter wise Coverage from Main Reference Book(s):

Unit No.	Text Books	Topics/Subtopics
I	Book-1	1.1 to 1.2, 2.1 to 2.6, 3.1 to 3.6, 26.1 to 26.4
II	Book-1	4.1 to 4.7,5.2 to 5.5, 6.1 to 6.4
III	Book-1	7.1 to 7.4, 8.1 to 8.5, 9.1 to 9.5,
IV	Book-1	10.1 to 10.4, 11.1 to 11.4, 12.2 to 12.3, 13.1 to 13.4
V	Book-1	14.1 to 14.4, 16.2 to 16.10,17.2 to 17.8, 19.1 to 19.6
VI	Book-1	21.1 to 21.6, 22.1 to 22.3

7. Accomplishments of the student after completing the course:

- Utilize the concepts in software development lifecycle.
- Demonstrate their capability to adopt quality standards.
- Assess the quality of software product.
- Apply the concepts in preparing the quality plan &documents.

Practical List

A. Setup: Using Java

I. Download and install the Java Software Development Kit(JDK) http://www.oracle.com/technetwork/java/javase/downloads/index.html
II. Download "Eclipse IDE for Java Developers" http://www.eclipse.org/downloads/
III. Download the Selenium Java Client Driver http://seleniumhq.org/download/
IV. Configure Eclipse IDE withWebDriver

Using Python

I. Download and install Python
II. Download and install Selenium Webdriver
III. Get Drivers for browser
IV. Install Pycharm (or any Python IDE)
V. Configure Pycharm IDE (or any Python IDE) with WebDriver

B. Suggested Applications for QA &QC:

1	**Leave Management System with following modules:** a. Login – Two types of User: Admin and User b. AdminFunctionalities: i. Manage LeaveTypes ii. Manage UserLeaves iii. Manage Users iv. Manage DifferentShifts v. Manage Reporting Groups and TeamStructure c. Time and Attendance User can view his/her attendancedetail Admin can view user's attendancelog Admin can generate various report like LateIn, EarlyOut, etc. d. Leaves User can apply leave and Admin can reject/approve User can view his leave request log, can modify and cancel as well
2	**Recruitment System** a. Manage Positions and vacancies within an organization b. Manage Advertisement on the positions c. Manage vacancy process from application to acceptance or rejection d. Manage and Setup Interviews e. Maintain Database
3	**Consider the below module which is responsible for Package purchase and Payment in different currencies:** • The User can purchase one out of three packages namely, Basic, Moderate, & Premium. • The User can purchase a particular package in three currencies. • The available currencies are Dollar, which is by default, Pound, &INR. • User can change the currency only on first step of 3, which is while selecting the package. a. The User Selects the Package. b. The User calculates the VAT amount applicable with respect to the country selection. c. The User pays online through debit/ credit card only. • List out all the possible Test Cases for above module. • List out critical step, which if not properly handled can deviate the user from the system. • List out the enhancement point which can avail user more flexibility about one of the above three steps
4	**In Airline reservation system, the following features need to be tested namely,** a. Login b. Search and book flights c. Search and book packages d. Register Feature not in scope, e. Search and book hotels • Pre-requites: Database & Payment gateway's sanboxenviroment access should

	be available. • Prepare the Test Plan for the above with all the possible criteria need Tobe considered. • Prepare the Test Cases for the features in scope to be tested. (At least one for each above mentioned feature) Prepare the Defect Report.
5	**Healthcare Web application with following modules:** a. Patient Registration b. Scheduling c. Treatment d. Billing
6	**Consider a School Management System,** which allows Parent to download Children's Progress Report and Results from School's website So How can we authenticate the User (parent)? &it has dual Authentication system integrated on Login page.
7	**Notepad "Save As" Functionality** What are the different scenarios that can be identified for testing a simple notepad save as functionality? The Save as does navigate to the file name and file path. Given these two fields what are the various scenario that can be applied for testing?

Part 1: Software Testing (Manual) - Mandatory

A. Create Test cases (Unit, Integration, System and Acceptance Test Cases) for Application
B. Perform manual testing using test case created and prepare test Metrics

Part 2: Suggested Template for Test case creation.

Sr#	Test condition / Steps	Input	Expected Result	Actual Result	Pass/Fail

Write test cases using following techniques (Suggested)

1. Line Coverage using Cyclomatic Complexity
2. Boundary Value Analysis(BVA)
3. Equivalence Partition(EP)
4. State Transition Technique v) Error Guessing Technique
   Test Case Management Tools like JIRA (desirable)

Part 3: Software Testing (Automated) - Mandatory

Test automation – script creation and execution

Tools: Selenium

A. Concepts

1	Introduction to Selenium, Installation and Setup

2	Selenium WebDriver Commands • BrowserCommands • NavigationCommands • WebElementCommands • FindElement and FindElementsCommand • CheckBox& Radio ButtonOperations • DropDown& Multiple SelectOperations • Handle Dynamic Web Tables in SeleniumWebdriver
3	Navigate back/forwards, get, refresh • I_ loading a page in current window / New window • Move back and forward • Refresh Page
4	Interrogation: get windowtitle currenturl Pagesource
5	Locating web elements by Id, ClassName, LinkText, PartialLinkText, Name, TagName, CssSelector ,XPath
6	Inspecting elements in web browsers
7	Element interrogation
8	Manipulation: Click, submit, shift-click, special actions, type text, clear text, list box selection and manipulation commands
9	Synchronization: Page load time out, implicit wait, explicit wait,ExpectedConditions class
10	Window handling: size, position, handles, switch to
11	Screenshot/capture
12	Browser profile: set preferred language, changing user agent, enable extension
13	Cookies: reading, creating and deleting
14	Data Driven Testing; Use pre-stored data as input and expected output (possible data Source file Excel)
15	Page Object Model: test scenario like shopping cart from login

B. Practical List

Write script and perform Following list of activities / test scenarios using offline and online web application (Permissible)

1	Open URL in different browsers (Chrome, Firefox, IE)
2	Perform mouse hovers and other events
3	To take snapshots.
4	Getting current time-stamp
5	To Handle a drop-down.
6	Handling 2 3 steps at a time like Website Registration+login+logout at a time in single run.

7	To Upload File.
8	Storing and fetching data from excel (E.g. Registration data)
9	Radio button selection.
10	Checkbox Selection
11	Selection
12	Absolute/Relative XPaths
13	Basic Selenium framework (Advance).
14	Generate Reports (Advance)
15	Logging (Advance).
16	Handling multiple browser tabs.
17	Handling multiple div./frames of page. (Advance)
18	Handling browser's parent and child window.(Advance)
19	verifies an expected page title, UI Element, Text and Table content
20	Handling mouse over menu/sub-menu.
21	To Provide meaningful messages in assertions!
22	Test Login of any web application
23	Test Shopping cart
24	Test using excel file data as input
25	Generate test report

References:

https://www.softwaretestinghelp.com/selenium-webdriver-commands-selenium-tutorial-17/http://toolsqa.com/selenium-webdriver/

Part4: Advanced Testing - (Desirable)

a. Perform SecurityTesting

1. Security testing for architecture, source code and user interface
2. Vulnerability testing with help of open source and licensed tools

References: www.owasp.org: OWASP guideline compliance verification

Type: Manual / Automated (using Tool) - (Desirable)

b. Perform Performance Testing

Performance testing for functionality for 50 concurrent users

Type: Manual / Automated (using Tool)

Part5: Issue Tracking(desirable)

Tools: JIRA, Bugzilla

Web References:

1. http://www.opensourcetesting.org/
2. http://www.onestoptesting.com/

Note: Some of the practical's form the above practical list may have seemingly similar definitions. For better learning and good practice, it is advised that students do maximum number of practical's. In the practical examination, the definition asked need not have the same wordings as given in the practical list. However, the definitions asked in the exams will be similar to the ones given in the practical list.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Big Data Tools
Subject Code: 629407

With effective from academic year 2020-21

1. **Learning Objectives:**
   - **a.** To understand basics of Big Data
   - **b.** To understand MongoDB, Hadoop, Map reduce, Pig and Hive

2. **Prerequisites:** Working knowledge of Programming Language and Database Concepts

3. **Contents:**

Unit	Course Content	Weightage percentage
Unit I	**Unit 1: Introduction to Big Data** Types of Digital Data: Classification of Data (Structured, Semi-structured and Unstructured), Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges of Big Data, Characteristics of Big Data (Volume, Velocity, Variety), Other characteristics of Big Data which are not Definitional Traits of Big Data, Why Big Data? Are we Information Consumer or Producer? Traditional BI vs Big Data, Typical Data Warehouse Environment, Typical Hadoop Environment, what is Changing in Realms of Big Data? Terminologies used in Big Data Environments	**15%**
Unit II	**Unit 2: Introduction to NoSQL and Hadoop** **NoSQL:** Introduction: Where is it used? What is it? Types of NoSQL databases, Why NoSQL?, Advantages of NoSQL, Use of NoSQL in Industry, SQL vs NoSQL, NewSQL **Hadoop:** Introduction, Distributed Computing Challenges, History of Hadoop, Overview of Hadoop and Hadoop Ecosystems, Features and key advantages of Hadoop, Versions of Hadoop, Hadoop distributions, RDBMS versus Hadoop, Hadoop vs SQL, Integrated Hadoop Systems offered by leading market vendors, Cloud based Hadoop solutions, HDFS, Processing data with Hadoop, Managing Resources and applications with Hadoop YARN, Interacting with Hadoop Ecosystem	**25%**
Unit III	**Unit 3: Introduction to MongoDB and Map Reduce** **MongoDB**: Introduction: What is MongoDB? Why Mongo DB? (using JSON, Creating or generating a unique key, Support for Dynamic Queries, Storing Binary Data, Replication, Shading, Updating information in –place), Terms used in RDBMS and Mongo DB, Data types in Mongo DB, MongoDB Query Language **Map Reduce:** Data Flow, Map, Shuffle, Sort, Reduce, Hadoop Streaming, mrjob, Installation, word count in mrjob, Executing mrjob	**25%**
Unit IV	**Unit 4: Introduction to HIVE and Pig** **HIVE:** Introduction: What is HIVE? HIVE Architecture, HIVE data Types, HIVE File Formats, HIVE Query Language, RCFile implementation, SerDe, User-Defined Functions (UDF)	**25%**

	Pig: Introduction: What is Pig? The anatomy of Pig, Pig on Hadoop, Pig philosophy, Use Case for Pig- ETL Processing, Pig Latin overview, Data types in Pig, Running Pig, Execution modes of Pig, HDFS commands, Relational operators, Eval function, Complex Data Types, Piggy Bank, User-defined Functions, Parameter substitution, Diagnostic Operator, Word Count Example using Pig, when to use and not use Pig? ,Pig at Yahoo, Pig vs HIVE.	
Unit V	**Unit 5: Overview of SPARK** Introduction to Data Analysis with Spark, Downloading Spark and Getting Started, Programming with RDDs	**10%**

4. Text Book(s):

1) Seema Acharya, Subhashini Chellappan, " Big Data and Analytics", Wiley India Pvt. Ltd.,2015
2) Matei Zaharia, Patrick Wendell, Andy Konwinski, Holden Karau, "Learning Spark", O'Reilly Media,2015
3) Zachary Radtka and Donald Miner, "Hadoop with Python'', O'Reilly Media,2016
(Free eBook is available on the following link) (As on 12-10-2018)
https://www.oreilly.com/programming/free/hadoop-with-python.csp

5. Reference Books:

1) Shashank Tiwari, "Professional NoSQL", Wiley India Pvt. Ltd.,2011
2) Kyle Banker,Peter Bakkum,Shaun Verch,Douglas Garrett,Tim Hawkins,"MongoDB in Action", DreamTech Press, 2nd Edition ,2016
3) Chris Eaton,Paul Zikopoulos,Tom Deutsch,George Lapis,Dirk Deroos,"Understanding Big Data : Analytics for Enterprise Class Hadoop and Streaming Data", Mcgraw Hill Education (India)Pvt.Ltd.,2012
4) Tom White,"Hadoop: The Definitive Guide",O'Reilly Media,4th Edition,2015
5) Vignesh Prajapati,"Big Data Analytics With R and Hadoop", Packt Pub Ltd ,2013
6) Services, "Big Data - Black Book", Dreamtech Press,2016

Web Resources:

1) http://www.bigdatauniversity.com
2) http://www.mongodb.com
3) http://hadoop.apache.org/

6. Unit wise coverage from Textbook(s):

Unit 1	**Book#**	**Topics**
I	1	Chapter. 1, 2, 3.12
II	1	Chapter 4,5
III	1,3	Chapter 6 (Book 1), Chapter 2 (Book 3)
IV	1	Chapter 9,10
V	2	Chapter 1,2 and 3 (For Chapter 2 and 3, only Python, No Java, No

		Scala)

7. Accomplishment

Upon completion of this course, students will be able to do the following:

- Students will learn difference between conventional SQL query language and NoSQL basic concepts
- Students will be able to design and build MongoDB based Big Data Applications and learn MongoDB query language
- Students will be able to write Map-Reduce based Applications

Practical List

Part I: Mongo DB

- Learn to Use MongoDB Atlas (The Cloud Version of MongoDB)
- Install and configure MongoDB

MongoDB Shell Commands / Queries: View all databases, Create new database, Drop existing database, View current database, Switch over to a given database, db. Help(), Display statistics of a given database, Display current version of MongoDB Server, Display list of collections in current database, Create Collection, Drop Collection, CRUD operations (Create, Read, Update, Delete), Insert, Update else insert, save, update, remove, Find, Dealing with Using NULL Values, Count, Limit, Sort, Skip, Arrays and Array Operations, Aggregate

1) Create a Student Master database with a collection called "Student" containing documents with some or all of the following fields: StudentRollNo, StudentName, Grade, Hobbies, and DOJ. Perform the following operations on the database:

a) Insert 10 Records in the database.
b) Find the document wherein the "StudName" has value "Ajay Rathod".
c) Find all documents in proper format. (Without _Id field)
d) Retrieve only Student Name and Grade.
e) Retrieve Student Name and Grade of student who is having _id column is 1.
f) Add new field "Address" in Student Collection.
g) Find those documents where the Grade is set to 'VII'.
h) Find those documents where the Grade is not set to 'VII'.
i) Find those documents where the Hobbies is set to either 'Chess'or is set to 'Dancing".
j) Find those documents where the Hobbies is set neither to 'Chess' nor is set to 'Dancing".
k) Find those documents where the student name begins with 'M'.
l) Find those documents where the student name has an "e" in any position.
m) Find those documents where the student name ends in "a".
n) Find total number of documents.
o) Find total the number of documents where Grade is 'VII'.
p) Sort the documents in ascending order of student name.
q) Display the last two records.

2) Create a MovieMaker Database with a collection called "Movies "containing documents with some or all of the following fields: titles, directors, years, actors. Perform the following operations on the database (either in the console or using any programming language):

a) Retrieve all documents
b) Retrieve all documents with Director set to "Quentin Tarantino"
c) Retrieve all documents where actors include "Brad Pitt".
d) Retrieve all movies released before the year 2000 or after 2010.
e) Add a synopsis to "The Hobbit: An Unexpected Journey": "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."
f) Add a synopsis to "The Hobbit: The Desolation of Smaug": "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
g) Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"
h) Find all movies that have a synopsis that contains the word "Bilbo"
i) Find all movies that have a synopsis that contains the word "Gandalf"
j) Find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"
k) Find all movies that have a synopsis that contains the word "dwarves" or "hobbit"
l) Find all movies that have a synopsis that contains the word "gold" and "dragon".
m) Delete the movie "Pee Wee Herman's Big Adventure"

3) Create a database named "BookStore" in MongoDB with a collection called "Books" containing documents with some or all of the following fields: bookId, bookTitle, authors (containing fields: authorName), publicationYear, publisher, Orders (containing fields: OrderedId, orderDate, customerName, price, quantityOrdered, discount).

4) Note that a book may have one or more authors and orders. Also, the same Ordered can be present in one or more books. Perform the following operations on the database (either in the console or using any programming language):

a) Insert records for 10 books from 5 authors, and at least 20 orders in total.
b) Update the title of a particular book.
c) Display all the books having less than 3 authors and sort by book name.
d) Display the number of books from each publisher.
e) Use Map Reduce function to display the total quantity of books ordered for each date.
f) Use Map Reduce function to display the discount offered to a particular customer.

5) Create a database named "Store" in MongoDB with a collection called "Sales" containing documents with some or all of the following fields: customerId, customerName, gender, dataOfBirth, contactNumber, address (containing fields: houseNo, street, area, city, pincode), orders (containing fields: orderId, orderDate, items (containing fields: itemId, itemName, itemPrice, quantityOrdered, discount)). Note that some customers may not provide their date of birth and/or contact number. Also, not all products would be sold at a discount. Perform the following operations on the database (either in the console or using any programming language):

a) Insert records for 3 customers and 5 items in at least 20 orders.
b) Update the contact number of a particular customer.
c) Display customerId, customerName, gender, contactNumber, of customers residing in "Ahmedabad".
d) Display city-wise count of customers
e) Use MapReduce function to display the number of times each item was sold.

6) Create a database "BookStore" with a collection called "Books" containing documents with some or all of the following fields: Category, BookName, Author, quantity, price, pages. Perform the following operations on the database:

a) Insert Records for 5 books.
b) Write Map & Reduce functions to split the books into the following two categories: Bigbooks, Smallbooks. (Books which have more than 300 pages should be in the Big books category. Books which have less than 300 pages should be in the Small books category.)
c) Count the number of books in each category
d) Store the output as follow as documents in a new collection called "Book Result".

Book Category	**Count of the Books**
Big books	2
Small books	3

Part II: Hadoop HDFS

- Installation and configuration for: Apache Hadoop Stand-Alone Mode and Pseudo
- Distributed Mode
- Installation and configuration for: Apache Hadoop Real Cluster consisting of a single
- Master and Two Slave nodes.
- Test the above set-up with sample examples bundled along with the downloaded package.
- To develop and execute sample programs like word-count, maximum temperature, etc. Using Python with Map-Reduce in Hadoop
- HDFS Commands: -ls, -ls -R, -mkdir, -put, -get, -copyFromLocal, -copyToLocal, -cat, -cp, -rm-r

1) Create a file "Sample" in a local file system and export it to the HDFS File System.
2) Write the HDFS command for copying a "Sample" file from HDFS to local File System.
3) Write HDFS commands for creating "Test" directory in HDFS and then removing that directory.
4) Write HDFS command to display complete list of directories and files of HDFS.
5) Write HDFS command for displaying the contents of "Sample" text file in HDFS on screen.
6) Write HDFS command for copying an existing "Sample" file in a "Test" HDFS directory to some another HDFS directory.

Part III: MapReduce

1) Prepare an "input" folder containing multiple text files. Create a program using MapReduce that would accept the path to the "input" folder and generate an "output" folder having a text file containing the total number of occurrences of each single word present in text document. For example, if the text containing in input files is as follows:
2) "We thank you for your visit to Ahmedabad. We hope that you would visit us again."
3) The Output should be as follow:

Word	Word Length	No of occurrences
We	2	2
To	2	1
Us	2	1
You	3	2
For	3	1
Your	4	1
That	4	1
Hope	4	1
Thank	5	1
Visit	5	2
Would	5	1
Again	5	1
Ahmedabad	9	1

1) Write a program for Matrix Vector Multiplication using MapReduce.
2) Write a program to perform Union, Intersection and Difference operation using
3) MapReduce on following files.

Input files:

1) Content of file 1 (apple, orange, mango, apple, banana)
2) Content of file 2 (apple, apple, plum, kiwi, kiwi, mango, mango)
3) Content of file 3 (orange, orange, plum, grapes, kiwi, mango, apple)

Part IV: Pig

- Install and configure Apache Pig
- Test the Pig Installation for local and map-reduce mode execution
- Test the Pig Installation for Interactive (Grunt Shell) and Batch Mode (. pig file) Execution
- Develop UDF (User Defined Function) in Python for Pig

Working with Pig Operators/Functions (LOAD, DUMP, FOREACH, GROUP, DISTINCT, LIMIT, ORDER BY, JOIN, UNION, SPLIT, SAMPLE, AVG, MAX, COUNT, TUPLE, MAP,PIGGY BANK, PARAMETER SUBSTITUTION, DESCRIBE, Simple Problems like Word Count using PIG

1. Write a pig script to load and store "Student data". (Student file contain Roll no, Name, Marks and GPA).
    a. Filter all the students who are having GPA>5.

b. Display the name of all Students in Uppercase.
c. Group tuples of students based on their GPA.
d. Remove duplicates tuple of Student list.
e. Display first three tuples from "student" relation.
f. Display the names of students in ascending order.
g. Join two relations namely Student and department (Rno, DeptNo, DeptName) based on the values contain in the roll no column.
h. Merge content of two relation Student and department.
i. Partition a relation based on the GPA's acquired by students.
j. To calculate the average marks for each student.
k. Calculate maximum marks of each student.
l. Count the number of tuples in a bag.

2. Load the file menu.csv (Category, Name, Price) and write one Pig script

a. Which meals cost more than 30.00?
b. Which meals contain the word "Panner"?
c. Which are the 10 most expensive meals?
d. For every day, what's the average price for a meal?
e. For every day, what's the most expensive meal?

- Write a program to count Word on Pig.
- Write a pig script to spilt customers for reward program based on their life time values.

If Life time values is >1000 and < =2000 then Silver Program
If Life time values is >20000 then Gold Program
Input:

Customers	Lifetime value
Jack	25000
Smith	8000
David	12000
John	15000
Scott	12000
Lucy	28000
Ajay	12000
Vinay	30000
Joseph	21000
Joshi	25000

1) Create a data file for below schemas:
Order: CustomerId, ItemId,ItemName,OrderDate,DelivaryDate
Customer: CustomerId,CustomerName, Address,City,State,Country

a. Load Order and Customer Data.
b. Write a pig latin Script to determine number of items bought by each customer.

- Do the Following:

1. Create a file which contains bag dataset as shown below.

User Id	From	To

user1001	user1001@sample.com	{(user003@sample.com),(user004@sample.com), (user006@sample.com)}
user1002	user1002@sample.com	{(user005@sample.com),(user006@sample.com)}
user1003	user1003@sample.com	{(user001@sample.com),(user005@sample.com)}

2. Write a pig latin statement to display the names of all users who have sent emails and also a list of the people that have sent the email to.
3. Store the result in a file.

7) Create a UDF to convert name into uppercase.

Part V: Hive

- Install and configure Apache Hive
- SerDe and User Defined Function Creation in Hive using Java

Create database, display list of existing databases, describe database, describe extended database, alter database properties, to make a given database as current database, drop database, create managed table, create external table, loading data into a table, working with collection data types, querying a table using select, querying collection data types, create static partition and load data into it from original table, static partition creation using alter, create dynamic partition, load data into dynamic partition, create bucket, create view, query view, drop view, sub-query, joins, Aggregation, Group By and Having, RC File Implementation

1. Create a data file for below schemas
Order: CustId,ItemId,ItemName,OrderDate,Delivary Date
Customer:CustId,CustName,Address,City,State,Country

1) Create a table for Orders and Customer Data.
2) Write Hive Query Language to find number of items bought by each customer.

2. Create a partition table for Customer Schema to reward customer based on their life time value.

Customer Id	Customers	Lifetime value
1001	Jack	25000
1002	Smith	8000
1003	David	12000
1004	John	15000
1005	Scott	12000
1006	Lucy	28000
1007	Ajay	12000
1008	Vinay	30000
1009	Joseph	21000
1010	Joshi	25000

1) Create partition table if life time value is 12000.
2) Create partition table for all life time values.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Java Web Technologies
Subject Code: 629408

With effective from academic year 2020-21

1. Learning Objectives:

- To learn and work with the web components of Java EE. i.e. the Servlet specification.
- Student will be able to learn MVC architecture and develop dynamic web application using Java Servlet and Java Server Pages technology.

2. Prerequisites: Programming Language of JAVA, HTML, JavaScript and JDBC

3. Contents:

Unit	Course Content	Weightage Percentage
Unit I	**Servlet Basics, Handling the Client Request: Form Data, HTTP Request Headers** Servlet Basics, Basic Servlet structure, Servlets Generating text/html content, Packaging Servlets, The servlet life-cycle. Handling Client Request Form Data, Reading Form Data from Servlets, Handling Client Request: Reading Request Headers, Understanding HTTP/1.1 Request Headers: Changing the page according to how the user got there and accessing the Standard CGI Variables.	**20%**
Unit II	**Server Response, HTTP Status Codes, HTTP Response Headers, Handling Cookies and Session Tracking** Specifying Status Codes, HTTP / 1.1 Status Codes, Using Redirections, HTTP Response Headers: Setting Response Headers from Servlets, Understanding HTTP / 1.1 Response Headers, Using Servlets to Generate JPEG Images, Handling Cookies: Remembering Usernames and Passwords, Deleting Cookies, Sending and Receiving Cookies, Using Cookie Attributes, Differentiating Session Cookies from Persistent Cookies, Using Cookies to Remember User Preferences, Session Tracking: Need for Session Tracking, Session Tracking API, Encoding URLs Sent to the Client and accumulating a List of User Data.	**20%**
Unit III	**Listeners and Filters** Using ServletContextListener, HttpSessionListener, Understanding of all the other Listeners viz. ServletRequestListener, ServletContextAttributeListener, ServletRequestAttributeListener, HttpSessionAttributeListener. Using Filters for pre and post processing of request.	**10%**
Unit IV	**Overview of Java Server Pages, Java Code with JSP Scripting, JSP Page Directives, Files and Applets in JSP and Java Beans Components in JSP** JSP Basic Syntax, HTML Text, HTML comments, Template Text, JSP Comment, JSPExpression, JSP Scriptlet, JSP Declaration, JSP 08 Directives, JSP Action, JSP Expression Language Element, Custom Tag (Custom Action), Escaped Template Text, Using JSP Scripting Elements, Using Predefined Variables, XML syntax for Expressions, Scriptlets, Declarations and Directives, Using Scriptlets, Using Declarations, Using Page Directive, Using Standard Actions Tags<jsp:plugin>, <jsp:forward>, <jsp: include>,	**20%**

	Using JavaBeans in JSP pages – <jsp:useBean>, <jsp:getProperty>, <jsp:setProperty>,Sharing Beans, Use of Scopes and their Attributes.	
Unit V	**Model-View-Controller (MVC), JSP 2.0 Expression and Accessing Database with JDBC** Integrating Servlets and JSP in a Web Application (MVC Architecture for Web Applications), Implementing MVC with Request Dispatcher, Understanding Data Sharing Between Servlets and JSP, JSP Expression Language, Accessing Scoped Variables, Bean Properties, Collections and Implicit Objects Using EL, Using EL Operators and Accessing Database with JDBC.	**20%**
Unit VI	**Declarative Security and Programmatic Security** Form-based authentication, Basic authentication, example, configuration Tomcat to use SSL, programmatic Security, Example, Handling all security programmatically, Example, Using programmatic security with SSL.	**10%**

4. Text Book(s):

- Marty Hall, Larry Brown, "Core Servlets and Java Server Pages Volume – 1", Pearson Education, 2nd ed. (2004)
- Marty Hall, Larry Brown, Yaakov Chaikin, "Core Servlets and Java Server Pages Volume – 2", Pearson Education, 2nd ed. (2004)

5. Reference Books:

- Black Book "Java server programming" J2EE, 1st ed., Dream Tech Publishers, 2008.
- Subrahmanyam Allamaraju, Cedric Buest, Professional Java Server Programming, Wiley Publication
- Pravin Jain, The Class of Java, Pearson
- Christian Bauer, Gavin King , Java Persistence with Hibernate, MANNING
- Giulio Zambon, Beginning JSP, JSF and Tomcat, Apress
- Cay S. Horstmann, "Core Java , Volume I – Fundamentals", Pearson Education , 10th Edition, 2017
- Cay S. Horstmann, "Core Java , Volume II – Advanced Features", Pearson Education , 10th Edition, 2017
- Cay Horstmann and Gary Cornell, Core Java, Volume II: Advanced Features, Pearson Publication
- James Keogh ,Complete Reference J2EE, McGraw-Hill publication

Web Resources

http://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html

6. Chapter wise Coverage from the Text Book:

Unit #	Book#	Chapter
I	1	Chapter 3 (except 3.5, 3.7, 3.8), Chapter 4 (only 4.2), Chapter 5 (only 5.1, 5.3, 5.6 and 5.7),
II	1	Chapter 6 (except 6.4), Chapter 7 (only 7.1, 7.2 and 7.5), Chapter 8

		(only 8.1, 8.3, 8.4, 8.6, 8.7, 8.11), Chapter 9 (only 9.1, 9.3, 9.5, 9.7)
III	2	Chapters 5, 6
IV	1	Chapters 10 (only 10.6), 11, 12, 13, 14
V	1	Chapters 15, 16, 17
VI	2	Chapters 3 (3.1 to 3.5), 4 (4.1 to 4.5)

7. Accomplishments

Students will understand advanced concepts related to MVC architecture, web services, servlet, spring and Hibernate. Students will be able to develop dynamic web applications using Java technology without need of other's help.

Practical List

1. Write a Servlet to display "Hello World" on browser.
2. Write a Servlet to display all the headers available from request.
3. Write a Servlet to display parameters available on request.
4. Write a Servlet to display all the attributes available from request and context.
5. Write a Servlet which displays a message and also displays how many times the message has been displayed (how many times the page has been visited).
6. Assume that we have got three pdf files for the MCA-1 Syllabus, MCA-2 Syllabus and MCA-3 Syllabus respectively, now write a Servlet which displays the appropriate PDF file to the client, by looking at a request parameter for the year (1, 2 or 3).
7. Assume that the information regarding the marks for all the subjects of a student in the last exam are available in a database, develop a Servlet which takes the enrollment number of a student as a request parameter and displays the marksheet for the student.
8. Develop a Servlet which looks for cookies for username and password, and forwards to a home.jsp in case the cookies are valid and forwards to login.jsp, in case the cookies are not found or the cookies are not valid.
9. Write a servlet to implement Session tracking using all four methods.
10. Develop a Servlet to authenticate a user, where the loginid and password are available as request parameters. In case the authentication is successful, it should setup a new session and store the user's information in the session before forwarding to home.jsp, which displays the user's information like full name, address, etc.
11. Write a simple JSP page to display a simple message (It may be a simple html page).
12. Write a JSP page, which uses the include directive to show its header and footer.
13. Create a listener that notifies (through System. Out) whenever a user adds a product to a shopping cart (i.e. adds an object to the session object) or removes it again. Hint: check out the class HttpSessionAttributeListener. Make it print the name and price of the object (hint: access the session through the HttpBindingEvent object). Also, let the listener print the total price of all objects saved in the session so far (one way to accomplish this could be to keep a collection of all objects saved to the session – or just their keys – in the listener or an associated class).
14. Create a servlet filter that logs all access to and from servlets in an application and prints the following to System. Out:
    1. the time the request was received
    1. the time the response was sent
    1. how much time it took to process the request

1. the URL of the resource requested
1. the IP address of the visitor

1) Develop an interest calculation application in which user will provide all information in HTML form and that will be processed by servlet and response will be generated back to the user.

2) Develop an application to demonstrate how the client (browser) can remember the last time it visited a page and displays the duration of time since its last visit. (Hint: use Cookie)

3) Develop an application to keep track of one user across several servlet invocations within the same browser session.

4) Develop an application to write a "page-composite" JSP that includes other pages or passes control to another page. (Hint: Use <jsp:include> or <jsp:forward>).

5) You want to reduce the amount of Java coding in your JSP using a JavaBean component. (Hint: Use <jsp:useBean> with the name of your bean).

6) Write a JSP page which uses tags available from the standard tag library JSTL.

7) Update the JSP page from above exercise to use tags available from the standard tag library JSTL.

8) Develop a JSP Page to display the personal information and result information of the student in two different tabular formats.

9) Create the filter that can add the time at which above jsp file called with appropriate message.

10) Design a Listener that loads the name of company as an init-parameter in context.
Use this name on the JSP – product.jsp and contactus.jsp of the website. (HINT: Implement ServletContextListener interface)

11) Create a filter to maintain the log of suspicious access of a particular JSP.
Access to the JSP on Sundays is suspicious.
Also design the JSP for the application. The JSP should display the details about sales history of the company for past 7 years.
The output should be shown in excel format.

12) Design an application where a user enters username and password, and requests for a servlet. Use filter to validate the password use database. If password is valid servlet is given as response, otherwise give appropriate message through filter.

13) Develop a program to perform the database driven operation like insert, Delete,
Update and select. To perform the above operations, create one table named Employee.

Field Name	**Field Type**
EmpId	Integer
Empname	Varchar
Emp_desig	Varchar
Emp_J_Date	Varchar
Emp_Salary	Numeric

14) Write a Java application to invoke a stored procedure using a CallableStatement. For this a stored procedure called increment Salary may be developed to increase all the employees salary by a percentage specified in the parameter.

15) Write a Servlet which uses the concept of Request forwarding & including external source in the current servlet context.

16) Write a JSP Page to which uses Session Tracking for online shopping.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Statistical Analysis using R
Subject Code: 629409

With effective from academic year 2020-21

1. **Learning Objectives:**
To understand and apply various concepts, techniques and methods used in Descriptive Statistics and Inferential Statistics. The knowledge and skills gained will equip students in carrying out preliminary Data Analytics tasks, and to prepare foundation to understand and apply the statistical techniques in various fields such as Total Quality Management, Simulation, Game Theory, Operations Research, etc. in addition to Computer Science topics such as Machine Learning, Cryptography, Artificial Intelligence, Operating Systems, Data Structures and Algorithms, etc.

2. **Prerequisites**: Preliminary mathematical concepts

3. **Contents**:

Unit	Course Content	Weightage Percentage
Unit I	**Introduction to Statistics and Descriptive Statistics** Introduction, Broad areas (classification) of Statistics; **Describing Data Visually:** Frequency Distributions and Histograms; Pie Charts; Bar Charts: Pareto Chart, Scatter Plots (Degree of Association); Line Charts; **Descriptive Statistics:** Central Tendency; Mean and its Characteristics, Median and its Characteristics, Quartiles and Percentiles, Mode; **Dispersion:** Range, Mean Absolute Deviation, Interquartile Range (IQR); Variance, Standard Deviation and its Characteristics, Coefficient of Variation; **Standardized Data:** including Chebyshev's Theorem, Outliers; **Box Plots:** including Fences and Unusual Data Values; **Grouped Data:** Nature, Mean and Standard Deviation, Accuracy Issues; **Skewness:** Coefficient of Skewness; **Kurtosis**: Leptokurtic, Platykurtic, Mesokurtic; **Measures of Association:** Covariance, Correlation, Coefficient of Correlation; Correlation and Causation	**18%**

Unit II	**Probability and Probability Distributions** **Introduction:** Common Framework: Experiment, Event, Elementary Events, Sample Space; Definition of Probability; Marginal Probability; Probability of Union of Events (Addition Laws), Probability Matrix; Probability of Complement of a Union; Probability of Joint Events (General Laws of Multiplication); Conditional Probability; Mutually Exclusive Events, Independent Events; Revision of Probability Values: Bayes' Rule **Discrete Probability Distributions:** Introduction, Binomial Distribution, Poisson Distribution, Applications; **Continuous Probability Distributions:** Introduction, Normal Distribution, Exponential Distribution, Applications;	**24%**
Unit III	**Sampling, Sampling Distributions and Estimation** **Types of Sampling:** Random, Nonrandom; Sampling Distribution of x-bar; Central Limit Theorem; z Formula for Sample Mean; Standard Error of Mean; Sampling from a Finite Population; Sampling Distribution of a Proportion, Standard Error of Proportion **Estimation for Single Population:** Estimating the Population Mean using z Statistic (σ Known); Estimating the Population Mean using the z Statistic when the Sample Size is Small; Estimating the Population Mean using t Statistic (σ Unknown); Estimating the Population Proportion; Estimating the Population Variance; Estimating Sample Size	**24%**
Unit IV	**One Sample Hypothesis Tests** Introduction; Null Hypothesis, Alternate Hypothesis; Type I & Type II Errors, Testing Hypotheses about Population Mean using z Statistic (σ Known); Using Critical Value Method to test Hypotheses, Examples; Testing Hypotheses about Population Mean using t Statistic (σ Unknown); Testing Hypotheses about Proportion; Testing Hypotheses about Variance	**18%**
Unit V	**Regression** Introduction, Simple Regression Analysis, Least Square Analysis to Determine the Equation of Regression Line; Residual Analysis, Using Residual to Test the Assumptions of the Regression Model; Standard Error of the Estimate; Coefficient of Determination; Hypothesis Testing for the Slope of the Regression Model; Testing the Overall Model; Using Regression to Develop a Forecasting Trend Line	**16%**

Optional (but Recommended) Topics:

- Overview of other Discrete and Continuous Probability Distributions
- Overview of Statistical Inferences about Two Populations; Analysis of Variance
- Overview of Multiple Regression Model; Mathematical Transformation of Nonlinear Models

4. Text Book:

1) Ken Black, "Business Statistics for Contemporary Decision Making", Wiley Student Edition, 2010

5. Reference Books:

1) David P. Doane, Lori E. Seward, "Applied Statistics in Business and Economics" Tata McGraw-Hill, 2010
2) Anderson, Sweeney, Williams, "Statistics for business and economics", 9th edition, 3**)** Thompson Publication
4) Bharat Jhunjhunwala, "Business Statistics", first edition, S Chand, 2008
5) Richard Levin, David Rubin, "Statistics for Management", 7th edition, PHI
6) Nabendu Pal, Sahadeb Sarkar, "Statistics-Concepts and Applications", 2nd edition, PHI
7) J. Susan Milton & Jesse Arnold, "Introduction to Probability & Statistics: Principles & Applications for Engineering & Computing Sciences", McGraw-Hill Education
8) S P Gupta, "Statistical Methods", 30th edition, S Chand

6. Chapter wise coverage from the Text Books:

Unit#	Chapter #
I	Chapter 1,2,3
II	Chapter 4,5,6
III	Chapter 7,8
IV	Chapter 9- (9.1-9.5) ,Chapter 10-10.1 , Chapter 11-11.1
V	Chapter 12 (12.2-12.7,12.9), Chapter :13 -13.1, Chapter 14-14.1

7. Accomplishment of the student after completing the course:

Students will be able to apply various concepts, techniques and methods used in Descriptive Statistics and Inferential Statistics in carrying out preliminary Data Analytics tasks. They will also be able to apply the statistical techniques in various fields such as Total Quality Management, Simulation, Game Theory, Operations Research, etc. in addition to Computer Science topics such as Machine Learning, Cryptography, Artificial Intelligence, Operating Systems, Data Structures and Algorithms, etc.

Practical List

Objectives: To implement statistical concepts using a standard tool, such as R. Such implementation is aimed at improved understanding and visualization of theoretical concepts. It is also aimed at laying a foundation for Data Analytics and Data Science.

Prerequisites: Logical Thinking and Basic Statistical Concepts

Advice (Note) to Teachers:
The list of exercises given below is an indicative list.

Note: R has many datasets. Get the available datasets through command data (). Use R commands related to Statistics for several datasets for a good practice.

Some exercises have been labelled as "**Mandatory**" while other exercises have been marked as "**Desirable**". It is expected that all the students will do **Mandatory** exercises while bright students will additionally do **Desirable** exercises as well.

List of Computer Lab Exercises

1. Introduction and a quick tour to R and R Studio (to be done in Lab) [09 Hours]

- **(a)** Basic data structures and constructs
- **(b)** Available R Datasets, such as mtcars, faithful, etc.
- **(c)** Null, NA, Missing Values
- **(d)** Basic Packages related to Statistics: e. g. stats, stats4, graphics, grDevices, modeest, agricolae, etc.

2. Descriptive Statistics [09 Hours]

- **(a)** Compute Mean, Median, Quartiles, Percentile (use quantile () function), Variance, Standard Deviation, IQR, Minimum & Maximum Values, Summary Statistics &interpretation(**Mandatory**)
- **(b)** Histogram, Scatter Plot, Box Plot, Density Plot of R data sets and interpretation (**Mandatory**)
- **(c)** Generate Frequency Distribution of data as a data frame(**Mandatory**)
- **(d)** Compute Correlation Coefficient and Covariance (**Mandatory**)

3. Probability and Probability Distributions [09 Hours]

- **(a)** Use pnorm(), pbinom(), ppois(), pexp() functions to compute probabilities based on a specific Probability distribution. (**Mandatory**)
- **(b)** Use dnorm(), dbinom(), dpois(), dexp() functions to compute probability density functions (**Mandatory**)
- **(c)** Use qnorm(), qbinom(), qpois, qexp() functions to get x value corresponding to given probability value (**Mandatory**)
- **(d)** Use different parameter values in 3 (a), and 3 (b) to observe the impact of different parameter values and prepare a note on that.(**Mandatory**)
- **(e)** Plot above results and interpret (**Desirable**)
- **(f)** Statistical test for normality using shapiro.test() function (**Desirable**)

4. Sampling, Sampling Distribution, Hypothesis Testing [12 Hours]

(a) Random sampling with or without replacement using sample () function (**Mandatory**)

(b) Generate n random samples (take n = 10, 50, 100, 200, 500, 1000 as an example), create a vector of Sample Means. Draw the Density Plot of Sample Means to visualize Central Limit Theorem (**Mandatory**)

(c) Take a sample and carry out Hypothesis Testing for the following cases:

1. Std. Deviation known, Large Sample Size, Sample from Non-Normal Population
2. Std. Deviation known, Small Sample Size, Sample from Normal Population
3. Std. Deviation known, Small Sample Size, Sample from non-Normal Population
4. Std. Deviation not known
5. Hypothesis Test for Variance (Chi-square Test) (**Desirable**)

5. Regression and Linear Modeling [06 Hours]

(a) Linear regression: One Independent Variable using lm () function; Interpret the output of Model Analysis, Compute Correlation Coefficient, Interpret results (**Mandatory**)

(b) Linear regression: Multiple Independent Variables using lm () function; Interpret the output of Model Analysis (**Mandatory**)

6. Using Tree data set (Inbuilt data set of R studio) create

(a) Histogram with proper label
(b) Scatterplot
(c) Boxplot
(d) Dot Plot
(e) Density Plot

7. Draw the different charts for Google Play Store data set (use following link)

https://docs.google.com/spreadsheets/d/12Uy4zsoUR44GfgVRtTIXRPr2DjwbW_-1rQN3lMextkg/edit?usp=sharing

(a) Rate wise Application
(b) Application popularity
(c) Number of users installing specific application
(d) Paid or Free Application
(e) Price wise Application
(f) Categorized Application (Entertainment, game, Education, Insurance etc.)
(g) Age group wise Application (Children, Teen, Mature, adult)

8. Perform Linear regression using employee dataset (use following link for dataset)
https://docs.google.com/spreadsheets/d/1r73YhZYxO1GiXZbiu1BREgJLlSyYhlv0y4jVLN2Niz4/edit?usp=sharing

Reference Books:

1. Pierre-Andre Cornillon, Arnaud Guyader, Francois Husson, Nicolas Jegou, Julie Josse, Maela Kloareg, Eric Matzner-Lober, Laurent Rouvière, "R for Statistics", CRC Press.
2. Dr. Mark Gardener, "Beginning R: The Statistical Programming Language", Wiley.
3. Paul Teetor, "R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics", O'Reilly Cookbooks.

Reference Websites:

1. https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
2. https://cran.r-project.org/web/packages/IPSUR/vignettes/IPSUR.pdf
3. https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/reading-questions-r-intro
4. https://www.datacamp.com/introduction-to-statistics
5. http://tut-dl.com/item/lynda-r-statistics-essential-training
6. https://www.analyticsvidhya.com › Machine Learning
7. https://www.coursera.org/learn/r-programming
8. https://www.analyticsvidhya.com/blog/2016/02/free-read-books-statistics-mathematicsdata-science/

Accomplishment of the student after completing the course:

1. Students will be able to carry out preliminary data analysis with results displayed graphically, and study the characteristics of standard probability distributions with their plots.
2. Students will also be able to demonstrate the inductive proof of Central Limit Theorem and go through linear regression (model) with fitness test of model.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Design and Analysis of Algorithms
Subject Code: 639401

With effective from academic year 2020-21

Prerequisites: Data Structure, C Programming Language.

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	4	5	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1	Analyze algorithms by working out for time and space complexity of algorithms, design algorithm and growth of functions.
CO2	Identify algorithm design methodology to solve problems. Analyze divide-and-conquer, probabilistic analysis and randomized algorithms.
CO3	Able to understand dynamic programming, its elements and analyze existing algorithm of dynamic programming.
CO4	To understand concept of greedy algorithms, amortized analysis and time complexity.
CO5	Apply knowledge of minimum spanning trees and design algorithm for shortest paths

3. Course Duration: The course duration is of 40 sessions of 60 minutes each.

4. Course Contents:

Module No:	Contents	No. of Sessions	70 Marks (External Exam)
Unit I	**Basic Concepts of Analysis and Design of Algorithms in Computing and Growth of Functions:** Algorithms, Algorithms as a technology, Analyzing algorithms, Designing algorithms, Asymptotic notation, Standard notations and common functions.	4	10
Unit II	**Algorithms Using Divide-and-Conquer, Probabilistic Analysis and Randomized Algorithms:** The maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences, The hiring problem, Indicator random variables, Randomized algorithms.	10	15
Unit III	**Dynamic Programming:** Rod cutting, Matrix-chain	6	12

	multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees.		
Unit IV	**Greedy Algorithms and Amortized Analysis** An activity-selection problem, Elements of the greedy strategy, Huffman codes. Aggregate analysis, The accounting method, The potential method and Dynamic tables.	10	15
Unit V	**Minimum Spanning Trees and Single-Source Shortest Paths*:*** Growing a minimum spanning tree, The algorithms of Kruskal and Prim, The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties.	10	18

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Text Book:

No.	Author	Name of the Book	Publisher
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	Introduction to Algorithms, Third Edition	MIT Press (2009).

8. Reference Books:

No.	Author	Name of the Book	Publisher
1	Parag H Dave, Himanshu B Dave,	Design and Analysis of Algorithms	Pearson (2014)
2	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	Introduction to Algorithms, 2nd Edition	PHI
3	Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran	Fundamentals of Computer Algorithms	Universities Press (2008)
4	Anany Levitin	Introduction to Design and Analysis of Algorithms	Pearson (2014)

5	S. Baase	Computer Algorithms: Introduction to Design and Analysis	Pearson (2002)
6	Aho, Hopcroft, and Ullman	The Design and Analysis of Computer Algorithms	Addison Wesley
7	Kleinberg	Algorithm Design	Pearson (2013)

Appendix-A

Sample Practical List

List of Mandatory Lab Exercises (Write Algorithms and Implement in C/C++ Language) for the following problems, students are expected to write one or more (as the case may be) algorithms along with the complexity of these algorithms, and implement them in C/C++ Language.

1. Find square root of a number. Can we use Divide & Conquer approach for this problem?
2. Determine smallest divisor of an integer.
3. For a given value of n, generate prime numbers <= n (more than one algorithms are possible)
4. Find Xn. Iterative and recursive algorithms are possible.
5. Determine product of 2 integers (a * b) as repeated sums. Iterative and recursive algorithms are possible.
6. Find Factorial of n. Iterative and recursive algorithms are possible.
7. Generate Fibonacci series up to n terms Iterative and recursive algorithms are possible.
8. Determine product of 2 large integers using multiplication of their digits. For simplicity, assume both numbers to have same number of digits. This assumption can be relaxed subsequently.
9. Program for finding maximum and minimum number using Divide and conquer.
10. Implement Recursive Binary search and Linear search and determine the time taken to search an element.
11. Breadth First Search (BFS) in a binary tree.
12. Depth First Search (DFS) in a binary tree.
13. Binary Search of an ordered array. Iterative and Recursive algorithms are possible.
14. Sort a given sequence of numbers using (a) Bubble Sort, and (b) Merge Sort
15. Generate permutations of given n numbers. Iterative and recursive algorithms are possible.
16. Find the closest pair out of given n points in 2-dimensional space.
17. Unique partitions of a positive integer.
18. Matrix multiplication using Dynamic Programming algorithm.
19. Solution of Rod-cutting problem using Dynamic Programming algorithm.
20. Generate pseudo-random numbers.
21. Implement Strassen’s algorithm for matrix multiplication
22. Prim's algorithm to find minimum cost tree (shortest path in a tree).
23. Kruskal's algorithm to find minimum cost tree (shortest path in a tree).
24. Implement Bellman-Ford Single Source Shortest Path Algorithm
25. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Machine Learning
Subject Code: 639402

With effective from academic year 2020-21

Prerequisites: Basics of computer science including algorithms, data structure, Probability theory and Bayesian Concept Learning *

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

1. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO-1	Understanding Various Machine Learning Algorithms & Their Area of Applications
CO-2	Explain the Process of Formulating & Solving Real World Problem using Machine Technology
CO-3	Design and Implement various Decision Making Problems using Machine Learning

2. Course Duration: The course duration is of **40 sessions of 60 minutes each.**

3. Course Contents:

Unit No:	Contents	No. of Sessions	70 Marks (External Exam)
I	• **Introduction to Machine Learning:** What is human learning? What is Machine Learning? Human learning versus machine learning, Types of machine learning, Applications of machine learning, Tools for machine learning.	4	7
II	• **Preparing to Model , Feature Engineering:** Machine Learning activities, Basic Types of data in Machine Learning, Structures of data, Data Quality and Remediation, Data Pre-Processing, Introduction to Feature Engineering, Feature Transformation, Feature Subset Selection	10	18
III	• **Modelling and Evaluation:** Selecting a Model, Training a Model, Model Representation and Interpretability, Evaluating Performance of a Model, Improving Performance of a Model	8	14
IV	• **Supervised Learning: Classification and Regression:** Example of Supervised Learning, Classification Model, Classification Learning Steps, Common Classification	10	17

	Algorithms Example of Regression, Common regression Algorithms		
V	• **Unsupervised Learning:** Unsupervised Learning versus Supervised Learning, Applications of Unsupervised Learning, Clustering, Finding Pattern using Association Rules	8	14

*** Additional Sessions may be taken to revise Prerequisites Contents.**

4. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical to be implemented using python library (Sample practical list is attached as Appendix-A)
- Interactive class room discussions

5. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

6. Reference Books:

No.	Author	Name of the Book	Publisher
1	Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das **	Machine Learning	Pearson Education
2	Tom M Mitchell	Machine Learning	McGraw Hill
3	Anuradha Srinivarasaraghavan, Vincy Joseph	Machine Learning	Wiley India
4	Peter Harrington	Machine Learning in Action	DreamTech
5	Henrik Brink, Joseph Richards, Mark Fetherolf	Real-World Machine Learning	DreamTech
6	Christopher Bishop	Pattern Recognition and Machine Learning	Springer
7	Jiawei Han and Michelline Kamber	Data Mining: Concepts and Techniques	Morgan Kaufmann

** The first Reference book is to be considered as Main Book as per Syllabus Contents

Appendix-A

Sample Practical List

1. Write a python code to implement **decision tree** for below given dataset. Identify the root node and all subpart or children of node and draw the tree.

Item no	Age	Income	Student	Credit Rating	Buys-Computer

1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	Middle	High	No	Fair	Yes
4	Senior	Medium	No	Fair	Yes
5	Senior	Low	Yes	Fair	Yes
6	Middle	Low	Yes	Excellent	No
7	Senior	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	Senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	Middle	Medium	No	Excellent	Yes
13	Middle	High	Yes	Fair	Yes
14	Senior	Medium	No	Excellent	No

2. Write a python code to implement **K-nearest neighbourhood** program for the given dataset.

Item no	Temp	Humidity	Wind Speed	Play
1	85	85	12	No
2	80	90	9	No
3	83	86	4	Yes
4	70	96	3	Yes
5	68	80	5	Yes
6	65	70	20	No
7	64	65	2	Yes
8	72	95	12	No
9	69	70	5	Yes
10	75	80	2	Yes
11	75	70	3	Yes
12	72	90	4	Yes
13	81	75	5	Yes
14	71	91	15	No

3. Write a python code to implement Apriori algorithm, apply join and prune method and find frequent item set

Sr No.	**Item no**	**Name**
1	T1	Bread, butter, milk, soda
2	T2	Coke, egg, milk
3	T3	Bread, butter, egg
4	T4	Break, coke, jam
5	T5	Bread, butter
6	T6	Potato chips, soda

7	T7	Coke, fruit, juice
8	T8	Bread, coke, milk
9	T9	Coke, soda, jam, milk
10	T10	Bread, butter, egg, milk, soda
11	T11	Bread, milk
12	T12	Bread, jam

4. Write a python code to apply **Naive Bayesian and Logistic Regression** algorithm to classify that whether a person can buy computer or not based on given test data:

Item no	**Age**	**Income**	**Student**	**Credit Rating**	**Buys-Computer**
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	Middle	High	No	Fair	Yes
4	Senior	Medium	No	Fair	Yes
5	Senior	Low	Yes	Fair	Yes
6	Middle	Low	Yes	Excellent	No
7	Senior	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	Senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	Middle	Medium	No	Excellent	Yes
13	Middle	High	Yes	Fair	Yes
14	Senior	Medium	No	Excellent	No

Test Data
Age: Youth Income: LOW Student: No Credit Rating: Fair Buy Computer -??

5. Python code for **Preparing to Model** (Appendix B.2, Page No. 365 – 373) and **Feature Engineering** (Appendix B.4, Page No. 378 – 385).

6. Apply k-means clustering approach with k = 2 to the following dataset.

X	Y	Z
-0.154	0.376	0.099
-0.103	0.476	-0.027
0.228	0.036	-0.251
0.330	0.013	-0.251
-0.114	0.482	0.014
0.295	0.084	-0.297
0.262	0.042	-0.304

-0.051	0.416	-0.306

7. Implement a python program that takes interest rate (x), finds the equation that best fits the data and is able to forecast out median home price for given interest rate using the data given below. (Use linear regression)

Interest rate (%)(x)	Median home price (y)
10.3	$183,800
10.3	$183,200
10.1	$174,900
9.3	$173,500
8.4	$172,900
7.3	$173,200
8.4	$173,200
7.9	$169,700
7.6	$174,500
7.6	$177,900
6.9	$188,100
7.4	$203,200
8.1	$230,200
7	$258,200
6.5	$309,800
5.8	$329,800

8. Apply following supervised machine learning algorithms for **All** classification problems as shown under:

Algorithm	Data set for Classification Problem
Decision Tree	Iris Dataset (from sklearn.dataset import load_iris)
Logistic Regression	Wine Dataset (from sklearn.dataset import load_wine)
K-Nearest Neighborhood	Breast Cancer Wisconsin Dataset (from sklearn.dataset import load_breast_cancer)
Naïve Bayes	Titanic Dataset (https://www.openml.org/d/42438)
SVM	Bank marketing Dataset (https://www.openml.org/d/1461)

Obtain model accuracy, generate classification report, and show the performance of the model graphically through roc_auc curve.

9. Build a prediction model using regression technique for (1) Boston house-prices (from sklearn.dataset import load_boston) (2) Diabetes (from sklearn.dataset import load diabetes) datasets. Also, evaluate the model.
10. Implement support vector machine approach to predictive modelling for (1) Boston house-prices (from sklearn.dataset import load_boston) (2) Diabetes (from sklearn.dataset import load diabetes) datasets. Also, evaluate the model.
11. Implement unsupervised machine learning algorithm (Clustering – K Means) in python on Titanic dataset (https://www.openml.org/d/42438) to cluster data by

removing the classlabel.Use elbow method to find the optimal number of clusters. Also, measure the quality of clustering via silhouette coefficient.

12. Implement unsupervised machine learning algorithm (Clustering – K Means) in python on Breast Tumour dataset to cluster data (use Breast Tumour dataset) by removing the classlabel.
13. Implement unsupervised machine learning algorithm (Clustering – Hierarchical) in python on Titanic dataset to cluster data (use Titanicdataset).
14. Implement Apriori algorithm in python to find rules which explain association between different products for given transactions at a retail store. (The data is available athttps://drive.google.com/file/d/1NUXoptUlHY8z4KcFKpFA6sQN5KnWzk3p/view?us p=sharing)

Desirable Practical Lists

1. Classification and Prediction algorithms on UCI dataset using Python's scikit-learn library. Perform Exploratory Data Analysis, implement data visualization techniques, pre-process the data, build a model by applying suitable algorithm, evaluate the model by generating report, and show the performance of the model graphically through roc_auc curve.
2. For the sentiment analysis dataset given in link: https://drive.google.com/file/d/1x6H7_KJjkbDrpgZFS7I2wjsZsILeSJ4S/view?usp=sharing, implement the following in python,
    a) Clean and pre-process the dataset by removing URL, removing HTML tags, handling negation words which are split into two parts, converting the words to lower cases, removing all non-letter characters
    b) Split the dataset into training and testing set
    c) Implement feature extraction technique (to convert textual data to the numeric form)
    d) Build the classification model using Logistic Regression that classifies if a given sentiment text is positive or negative
    e) Obtain the accuracy score of the built model.
3. Implement a content-based recommender system in python that recommends movies that are similar to a particular movie using movielens-20m-dataset available at https://kaggle.com.

The practical exercises should be performed in python.

References:

1) Peter Harrington, “Machine Learning in Action”,DreamTech
2) Michael Bowles, “Machine Learning in Python”,Wiley
3) Gavin Hackeling, Mastering Machine Learning with scikit-learn,Packt
4) Giuseppe Bonaccorso, Machine Learning Algorithms - Second Edition,Packt

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Software Engineering
Subject Code: 639403

With effective from academic year 2020-21

Prerequisites: Systems & Object Oriented Design Methodologies

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
4	-	-	4	70	30	-	-	100

1. Teaching and Examination Scheme:

2. Course Outcomes:

Course Outcome Component	Course Outcome(Learner will be able to)
CO1	To understand the concepts of software engineering, software process model.
CO2	Able to select and apply appropriate process model to all stages of software development life cycle (SDLC), requirements engineering and how to manage user's requirement
CO3	To know design concepts and user interface
CO4	To understand agile methodology and scrum.
CO5	Understand high level design and UML Diagram, develop prototype model for a given case study using modern engineering tools and students would be able to build an SRS documents of the project.

3. **Course Duration:** The course duration is of 40 sessions of 60 minutes each.

4. Course Contents:

Module No:	Contents	No. of Sessions	70 Marks (External Exam)
I	**Introduction to Software Engineering & Process Models** Software Engineering, Software Process, Process Models – Waterfall, Incremental, Evolutionary Process Model – Prototype, Spiral and concurrent Development Model, Agile Process; Extreme Programming (XP); Brief Overview of Other Agile Process Models: Adaptive Software Development and Scrum	**4**	**05**
II	**Requirement Engineering** Requirements Engineering; Groundwork for Understanding of Software, Requirements; Overview of Eliciting Requirements, Developing Use Cases, Building the Requirements Model; Negotiating Requirements; Validating Requirements;	**8**	**10**

	Requirement Modelling Strategies; Overview of Flow Oriented Modelling, Behavioral Modelling;		
III	**Design Concepts** Design Concepts, Design Model; Architectural Styles, Architectural Design, Assessing Alternative architectural Designs, Architectural mapping Using Data Flow, User Interface Design: Golden Rules of User Interface Design; User Interface Analysis and Design; Interface Analysis; Interface Design steps	**8**	**15**
IV	**Introduction to Agile Methodology** Agile Principles: 12 principles of Agile software, The customer is always right, Delivering the project, Communicating and working together, Project execution - Moving the project Along, Constantly Improving the Project and the Team, Agile Project: Bringing all the principles Together. Scrum and Self organizing Teams: The rules of Scrum, Everyone on a Scrum Team Owns Project, The whole team uses the daily Scrum, Sprints, planning and retrospectives, Scrum Planning and collective commitment: User stories, Velocity and Accepted Scrum Practices, Scrum Values revisited.	**10**	**20**
V	**HIGH LEVEL DESIGN** **Overview:** What to specify: Security, Hardware, User Interface, Internal Interfaces, External Interfaces, Architecture, Reports, Other Outputs, Database (Audit trails, User Access, Database Maintenance), Configuration Data, Data Flows and States, Training, UML Diagrams (Structure Diagram, Behavior Diagrams (Use case, Activity, State Diagram), Interaction Diagrams, Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram**.**	**10**	**20**

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Mid-Semester examination**	**(30 Marks)**
B	**End –Semester Examination**	**(70 Marks)**

7. Text Book:

No.	Author	Name of the Book	Publisher
1	Roger S. Pressman	Software Engineering – A Practitioner's Approach",	McGraw Hill Publications

		7th Edition	
2	Andrew Stellman, Greene Jennifer	Beginning Agile	Beginning Agile, O'Reilly
3	Rods Stephen	Beginning Software Engineering	WROX

8. Reference Books:

No.	Author	Name of the Book	Publisher
1	Sommerville	Software Engineering", 8th Edition	Pearson Education
2	Chandramouli Subramanian, Saikat Dutt, Chandramouli Seetharaman, B G Geetha	Software Engineering	Pearson
3	Waman S. Jawadekar,	Software Engineering– Principles and Practices	TMGH Publication
4	Pankaj Jalote	Software Engineering -A Precise Approach	Wiley India
5	Waman S. Jawadekar	Software Engineering - A Primer	TMGH Publication

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rdSemester
Subject Name: Software Project-2
Subject Code: 639404

With effective from academic year 2021-22

1. **Prerequisites:** Programming Language, Basic Concepts of Software Engineering, RDBMS.

2. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
-	-	4	2	-	-	80	20	100

3. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO-1	Develop Project using Emerging Framework
CO-2	Analyze real word issues and develop roadmap to resolve those issues using cutting edge technologies
CO-3	Explore Project Development Life Cycle

4. Guidelines for Project using Framework

- It is suggested to develop Web/Desktop/Mobile applications using emerging frameworks like Magneto, CodeIgniter, Flutter, React Native, Angular, Node.js or any other open source frame works.
- It is recommended that the team should be of 2-3students.
- The project should be free from plagiarism of any kind.
- Project must have proper documentation.
- This may or may not be a live project.
- Coding standards should be followed meticulously. At the minimum, the code should be self-documented, modular, and should use the meaningful naming convention.
- It is advisable that object-oriented methodology is used with reusability of classes and code, etc.
- The output reports must include MIS reports, if applicable.
- The documentation should include a chapter on “Learning during Project Work”, i.e. “Experience of Journey during Project Duration”.
- Student may be asked to write the code related to the project during examination.
- Mentor/ Internal guides (i.e. the faculty members) must devote time, allocated as per the time table to guide the students for the project. The time allocation will be in accordance with the teaching scheme for 5th semester project.

5. Documentation:

- The project has to be well-documented in the form of a Project Report (at least 50 pages comprising of the relevant description of the project including design, data dictionary, source code, screenshots, etc.).
- Format: The Project report Print out should be taken on both the side of page with single line spacing. Use Times New Roman of size 10 for normal text. A typical Table of content will be as follows.

TABLE OF CONTENTS

1. Introduction
    1.1. Existing System
    1.2. Need for the New System

1.3. Objective of the New System
1.4. Problem Definition
1.5. Core Components
1.6. Project Profile
1.7. Assumptions and Constraints
1.8. Advantages and Limitations of the Proposed System

2. Requirement Determination & Analysis
   2.1. Requirement Determination
   2.2. Targeted Users
3. System Design
   3.1. Use Case Diagram
   3.2. Class Diagram
   3.3. Interaction Diagram
   3.4. Activity Diagram
   3.5. Data Dictionary
4. Development
   4.1. Coding Standards
5. Agile Documentation
   5.1 Agile Project Charter
   5.2 Agile Roadmap / Schedule
   5.3 Agile Project Plan
   5.4 Agile User Story (Minimum 3 Tasks)
   5.5 Agile Release Plan
   5.6 Agile Sprint Backlog
   5.7 Agile Test Plan
   5.8 Earned-value and burn charts
6. Proposed Enhancements
7. Conclusion
8. Bibliography

6. Knowledge about the following is expected to be demonstrated.

- The objective of the Project Development is to make students aware about the industry based process and workings using Framework. As a result, Project must meet with the industry standards.
- Proper knowledge about the purpose of the application.
- Use of justifiable application for group of 2-3members.
- Project must include features like MIS Reports, Advance Search, File based processing etc.

7. Evaluation

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total – 20 Marks)**
	• Continuous Evaluation Component	10marks
	• Class Presence & Participation & Project Report	10marks
B	**End –Semester Examination(Practical/Viva)**	**(80 Marks)**

8. Assessment Weightage

Sr. No	Particulars	Marks
1	Project	24
2	Documentation	16
3	Code Changes	20
4	VIVA	20

Web References:

1) https://magento.com/
2) https://reactnative.dev/docs/getting-started
3) https://codeigniter.com/download
4) https://angular.io/docs
5) https://nodejs.org/en/download/
6) https://flutter.dev/docs/get-started/install
7) ionicframework.com/docs
8) https://visualstudio.microsoft.com/xamarin/
9) https://github.com/xamarin
10) https://www.djangoproject.com/

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Open Stack Technology
Subject Code: 639405

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
Virtualization	• Learner can get a brief idea about what is virtualization and various concepts of virtualization
OpenStack Concepts	• Learner will get to learn all the necessary concepts of OpenStack and OpenStack components
OpenStack Deployment	• Learner will get to learn basics of DevOps, and will learn the deployment of OpenStack
OpenStack Cluster and Compute	• Learner will get to learn about the concepts of cluster and would learn how to compute in OpenStack
Operating the OpenStack Infrastructure	• Learner will learn types of storages in OpenStack and various types of network deployment and would learn to operate the OpenStack Infrastructure

3. Course Duration: The course duration is of **40 sessions of 60 minutes each.**

4. Course Contents:

Module No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	• **Introduction** Introduction to Linux, Linux Commands, Virtualization Techniques, Various types of virtualization, Introduction to Cloud Computer, Characteristic of cloud computing, cloud computing models : Service Model and Deployment Model, Privacy and Security in Cloud Storage Services, What is Hypervisor, Xen, ESXi, Hyper-V, Understanding Docker	5	
II	• **Basics of OpenStack** Introduction to OpenStack and It's component, Other cloud frameworks, Comparing it to AWS and Rackspace Cloud, Creating a safe environment, Access Control, The CIA Model, Understanding Data Center Security from Cloud Perspective, Understanding Server Security from Cloud Perspective, Nova-Compute Service, Nova-API, Nova-Compute, Nova-Network, Nova-Scheduler, Nova-Conductor, Storages in Openstack	6	
III	• **OpenStack Deployment**	9	

	Understanding DevOps and OpenStack deployment, Working with the infrastructure deployment code, Integrating OpenStack into infrastructure code, Choosing the automation tool : Introducing Ansible, Ansible Modules, Ansible Variables, Ansible Inventory, Ansible Roles, Ansible Playbooks, Ansible for OpenStack, The development and production environments, Preparing the infrastructure code environment : Preparing the development Setup, Configuring the setup, Building the development setup		
IV	• **OpenStack Cluster and Compute** Understanding clustering : Asymmetric cluserting, symmetric clustering, The cloud controllers:The keystone service, the nova-conductor service, the nova-scheduler service, The API services, The network service, Cloud Controller Clustering : Deployment with Openstack Ansible, Bringing up the controller nodes, The Compute Service Components, Deciding on the hypervisor, Segregating the compute cloud, Overcommitment considerations, Storing instances alternatives, Understanding instance booting, Planning for service recovery	10	
V	• **Operating the OpenStack Infrastructure** Understanding the storage types, Deploying Swift service, Using block storage service: Cinder, Understanding Architecture of Neutron, Implementing Virtual Networks, Connecting Virtual Networks with routers, Implementing network security in OpenStack, Operating the OpenStack tenancy, Orchestration in OpenStack, Telemetry in OpenStack, Installing Telemetry, Arming OpenStack Monitoring	10	

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Reference Books:

No.	Author	Name of the Book	Publisher
1	Chandan Dutta Chowdhury and Omar Khedher	Mastering OpenStack	PacktPub
2	Kevin Jackson	OpenStack Cloud Computing Cookbook	PacktPub
3	Alessandro Locati Fabio	Openstack Cloud Security	PacktPub
4	John R. Vacca	Cloud Computing Security	CRC Press
5	Ronald L. Krutz, Russel Dean vines	Cloud Security: A comprehensive guide to secure cloud computing	Wiley

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Operations Research
Subject Code: 639406

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1	• Understand the various optimization models and their areas of application.
CO2	• Explain the process of formulating and solving real world problems using optimization methods.
CO3	• Describe different decision-making environments and apply decision making process in the real-world situations.

3. Course Duration: The course duration is of 45 sessions of 60 minutes each**.**

4. Course Contents:

Unit No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	• **Basics of Operations Research and Linear Programming** (i) Basics of Operation Research Operation Research introduction, definitions, features, advantages and applications (ii) Linear Programming Problem (L.P.P.) Linear Programming Problem (L.P.P.), Mathematical definition of a L.P.P. with its components: objective function and constraints, optimal solution, slack, surplus and artificial variables, Graphic method, Simplex method, Big – M method, Primal & Dual problem definition	12	18
II	• **Special Cases of L.P.P.** **(i) Transportation problem (T.P.)** Mathematical definition of a T.P., Method to find initial basic feasible solution, NorthWest corner rule, Least cost cell entry method, Vogel's approximation method, Test of optimality for finding an optimum solution – MODI method. **(ii) Assignment problem (A.P.)** Mathematical definition of an Assignment Problem, Method to find an optimum solution - Hungarian Method, Variations of assignment problem	10	14
III	• **Management of Inventory and Replacement** **(a) Management of Inventory**	08	14

	Introduction and terminology of the inventory management problem including Objective(s) and Constraints; Single Item Inventory Control without Shortages Model –I: EOQ model with constant rate of demand Model – II: EOQ model with different rates of demand **(b) Management of Replacement** Definition, replacement of items that deteriorates, replacement of item that fails completely		
IV	• **Project Management and Scheduling** **(i) Project Management (CPM & PERT)** Network concepts, components, rules for network construction, critical path method (CPM) and Project evaluation and Review Techniques (PERT) **(ii) Production scheduling (job sequencing)** Introduction, Johnson's algorithm for n jobs 2 machines, Johnson's algorithm for N jobs m machines, 2 jobs m machines using graphical method.	10	14
V	• **Queuing Theory** Introduction, Queuing system and problem, transient and steady states, traffic intensity, probability distributions in queuing systems, single service queuing model(s).	05	10

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment: Practical to be implemented using Python Based google or tools library or 'R' (Sample practical list is attached in Appendix-A)
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Reference Books:

No.	Author	Name of the Book	Publisher
1.	J. K. Sharma	Operations Research – Theory and Application*	Macmillan Publishers India

			Ltd.
2.	Kanti Swarup, Gupta P.K. , Man Mohan	Operations Research	Sultan Chand & Sons, New Delhi
3.	V. K. Kapoor	Operations Research – Problems & Solutions	Sultan Chand & Sons, New Delhi
4.	Shah, Gor, Soni	Operations Research	PHI

* Book mentioned at Sr no. 1 is considered as textbook to cover the topics.

Chapter wise coverage from Book 1.

Unit No	Topics	Chapter
1	Basics of Operation Research	Ch – 1 (1.1 to 1.5, 1.10, 1.13)
	Linear Programming Problem	Ch – 2 (2.1, 2.2,2.3,2.4, 2.6,2.7, 2.8)
	Simplex and Big-M Methods of LPP	Ch. 4 (4.1 to 4.6)
	Duality in LPP	Ch.5 (5.1, 5.2)
2	Transportation Problem	Ch – 9 (9.1 to 9.5)
	Assignment Problem	Ch – 10 (10.1 to 10.4)
3	Inventory Management	Ch – 14 (14.1 to 14.7)
	Replacement	Ch – 17 (17.1 to 17.4)
4	Project Management (CPM and PERT	Ch – 13 (13.1 to 13.6)
	Job Sequencing	Ch – 20 (20.1 to 20.3, 20.5, 20.6)
5	Theory of Queues	Ch – 16 (16.1 to 16.6)

Appendix-A
Sample Practical List

1. A toy manufacture produces two types of dolls; a basic version doll A and a deluxe version doll B. Each doll of type B takes twice as long to produces as one doll of type A. The company has time to make a maximum of 2,000 dolls of each type per day, and the supply of plastic is sufficient to produce 1,500 dolls per day and each type requires equal amount of it. The deluxe version, i.e. type B requires a fancy dress of which there are only 600 per day available. If the company makes a profit of Rs. 30 and Rs. 50 per doll, respectively, on doll A and B; how many of each should be produced per day in order to maximize profit? Formulate the problem as LPP and solve it by graphic method.

2. Solve using the Simplex method the following problem:
$$\text{Maximize } Z = f(x,y) = 3x + 2y$$
$$\text{Subject to: } 2x + y \leq 18$$
$$2x + 3y \leq 42$$
$$3x + y \leq 24$$
$$x \geq 0 , y \geq 0$$

3. Solve using the Simplex method the following problem:
Maximize $p = 2x - 3y + z$ Objective function subject to $x + y + z \leq 10$
$$4x - 3y + z \leq 3 \quad 2x + y - z \leq 10$$
$$x \geq 0, y \geq 0, z \geq 0$$

4. Develop a generalized program to solve optimized **Transportation problem**. First develop the program for a balanced problem, make a copy of that program and then modify to take care of unbalanced problem. Ask number of sources and destinations and the costs of transportation from every source to every destination. Show allocation at every step, final allocation and total transportation cost.

5. Develop a generalized program to solve optimized **Assignment problem**. First develop the program for a balanced problem, make a copy of that program and then modify to take care of unbalanced problem. Ask number of sources and destinations and the costs of transportation from every source to every destination. Show allocation at every step, final allocation and total transportation cost.

6. A certain item costs Rs. 75 per ton. The requirement is 8,000 tons per year and each time the stock is replenished there is a set – up cost of Rs. 600. The cost of carrying inventory has been estimated at 12.8 per cent of the value of the stock per year. Find out the optimal order quantity, number of orders required to be placed in a year, number of days between two successive orders and total variable inventory cost. Assume 360 days in a year.

7. A television repairman finds that the time spent on repairing each TV has an exponential distribution with a mean of 15 minutes. He repairs the sets in the order in which they arrive. The arrival of sets follows a Poisson distribution approximately with an average rate of 16 per 8 hour day. Find out for how many hours would the repairman be busy in a day, what is the average number of TV sets in the system and the average waiting time of a TV set in the system.

8. There are 5 workers and their work time to complete their jobs on different machines are given below. Develop a program to solve **Assignment problem** for minimum solution

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Worker 1	8	5	7	7	8
Worker 2	9	5	6	7	8
Worker 3	6	8	5	6	9
Worker 4	8	10	7	6	5
Worker 5	4	6	5	6	4

9. There are 5 salesman and each of them can work on any one of 5 districts. Table below shows average revenue generated by each of them. Develop a program to solve **Assignment problem** for maximization.

	District 1	District 2	District 3	District 4	District 5
Salesman 1	250	198	206	220	210
Salesman 2	240	220	196	208	212
Salesman 3	260	240	198	220	220
Salesman 4	240	250	194	208	200
Salesman 5	240	220	198	200	204

10. A television repairman finds that the time spent on his jobs has an exponential distribution with mean of 30 minutes. If he repairs sets in the order in which they came in, and if the arrival of sets follows a Poisson distribution approximately with an average rate of 10 per 8-hour day, what is the repairman's expected idle time each day? How many jobs are ahead of the average set just brought in?

11. On an average 96 patients per 24-hour day require the service of an emergency clinic. Also on an average, a patient requires 10 minutes of active attention. Assume that the facility can handle only one emergency at a time. Suppose that it costs the clinic Rs 100 per patient treated to obtain an average servicing time of 10 minutes, and that each minutes of decrease in this average time would cost Rs. 10 per patient treated. How much would have to be budgeted by the clinic to decrease the average size of the queue from one and one-third patients to half patient.

12. Students arrive at the head office according to a Poisson input process with a mean rate of 40 per hour. The time required to serve a student has an exponential distribution with a mean of 50 per hour. Assume that the students are served by a single individual, find the average waiting time of a student.

13. Develop a program to Find Critical Path, completion time, float time for following activity table.

Activity	Duration
1-2	6
1-3	8
2-4	3
2-5	5
3-5	9
4-5	6
5-6	8

Desirable:

14. Develop a generalized **sequencing** program for n jobs and m machines. First develop a program for n jobs two machines, make a copy and then make it general for n jobs m machines. Show the sequence after every iteration, final sequence, total elapsed time and idle times for every machine

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Cloud Computing
Subject Code: 639407

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1	• Understand concepts of Cloud Computing
CO2	• Understand virtualization techniques and Service Level Agreement
CO3	• Estimate cost for Cloud Computing implementation
CO4	• Understand and analyse architecture of cloud computing
CO5	• Deploying application in cloud environment

3. Course Duration: The course duration is of 40 sessions of 60 minutes each.

4. Course Contents:

Unit No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	• **Overview of Distributed Computing**: Computing, Traditional Utilities, Creation of the Internet, computing Paradigm Trends, Computing Paradigm Evolution, cloud computing: A New Paradigm, Differences and Similarities Among different types of computing. • **Introduction to Cloud Computing:** Definition, Central Ideas Behind Cloud Computing, Properties and Characteristics of Cloud Computing, Benefits of Cloud Computing, Cloud Service and Deployment Models, Organizational Scenario of Cloud: Cloud Deployment Model, Cloud Architecture, Cloud Vocabulary, Challenges with Cloud Computing, Cloud Supporting Services, Management and Administration of Cloud Services.	6	10
II	• **Virtualization Techniques:** Virtualization Technology, Overview of X86 Virtualization, Types of virtualization, Virtualization products, Concept of VLAN and Benefits, Concept of SAN and Benefits, VM Migration, VM Consolidation and Management, Cloud Interoperability Standards. • **SLA with Cloud Service Providers:** The concept of	6	10

	SLA, SLA Aspects and Requirements, Service Availability, Cloud Outages, Credit calculation for SLA Breaches, Sample SLA for Amazon, Rackspace, Google, HP etc.		
III	• **Risk, Consequences and costs for cloud computing:** Introducing Risks in cloud computing, Risk Assessment and Management, Risk of vendor lock-in, loss of control, not meeting regulatory compliance, resource scarcity or poor provisioning, Multi-tenant Environment, failure, supply chain, In adequate SLA, malware and Internet attack, Management of Cloud Resources, Network Outage, Physical Infrastructure, Legal Risks due to Legislation, Risk with Software and Application Licensing, Security and compliance requirements for public cloud, Calculating total cost of ownership(TCO) for cloud computing, direct and indirect Cloud costs, Cost allocation in the cloud, Chargeback model for Allocation of Direct and indirect costs, Chargeback Methodology, cost, Billable Items, Atomic Unit, Pricing Model, Charge back tools and solutions, maintaining Strategic Flexibility in a cloud.	6	10
IV	• **Application Architecture for cloud:** Cloud Application Requirement, Architecture for Traditional versus Cloud Applications, Assumptions for Traditional and Cloud Applications, Recommendation for cloud Application Architecture, Fundamental Requirements for cloud application Architecture, Relevance and Use of Client-server Architecture for cloud Applications, Addressing Cloud Application Performance and Scalability, Service Oriented Architecture(SOA) for cloud Applications.	6	10
V	• **Introduction to Google App Engine:** The Runtime Environment, The Static file servers, The Data store, The Services, Google Accounts, Task Queues and cron jobs, Developer tools, Administration console. • **Creating an Application:** Setting the SDK, Developing an application, registering the application, deploying the Application, and Uploading the Application. **Handling Web Requests:** The App Engine Architecture, Configuring the Front end, How App Runs, Quotas and Limits.	8	15
VI	• **Data Store Entities**: Entities, keys and Properties, Data Store API, Property values, Keys and Key Objects, Using Entities, **Persistence API:** Setting Up JPA, Entities and Keys, Entity Properties, Embedded Objects, Saving, Fetching, and Deleting Objects, Transactions in JPA, Queries and JPQL Relationships.	8	15

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Text Books:

No.	Author	Name of the Book	Publisher
1	Rishabh Sharma	Cloud Computing Fundamentals, Industry Approach and Trends	Wiley Publication.(ISBN: 978-81-265-5306-8)
2	Kailash Jayaswal, Jagannath Kallakurchi, Donald J Houde, Dr. Deven Shah	Cloud Computing: Black Book	Dreamtech Publications (ISBN 978-93-5119-418-7)
3	Dan Sanderson	Programming Google App Engine	O'Reilly | Google Press: (ISBN-978-0-596-52272-8)

8. Reference Books:

No.	Author	Name of the Book	Publisher
1	Anthony T. Vetle	Cloud Computing: A practical approach	Tata McGraw Hill Education Private Limited (2009)
2	Rajkumar Buyya, Christian Vechhiola, S. ThamaraiSelvi	"Mastering Cloud Computing"	McGraw Hill Education(India) Private Limited.
3	Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper	Cloud Computing For Dummies	Wiley India Pvt Ltd
4	Kris Jamsa	Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More (Student Edition)	Jones & Bartlett Learning

9. Suggestive Practical List:

1. Develop a hello world program web application and deploy it on the Google app engine.

2. Develop a web application which displays the current date and time in a formatted way.
3. Develop a web application which is customized version of the practical no 2, which relates the clock with Google accounts. Each user will get different view based on the preferences and the user's time zone.
4. Develop a web application for task management. Create a form to enter summary, URL and description of task and a button to insert a task in the task list. Use JPA (Java Persistence API) for the persistence and Sends the selected todo to yourself via email as a reminder
5. Develop a simple Java web application for Google App Engine--a guestbook that lets users post messages to a public message board.
6. Create a book class as entity and make the object persistence using JPA. Demonstrate the saving, fetching and deleting objects operations.
7. Create an employee class as entity and makes the objects persistence using JPA. Demonstrate transaction processing in the data store using JPA.
8. JPA includes a SQL-like query language called JPQL, perform the practical no 6 with JPQL also.
9. Perform the practical no 7 with JPQL[list the employees having salary between a particular range]

Note: Internet connection is required for each practical except practical 1 and practical 2

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Embedded System Design
Subject Code: 639408

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1: Comprehension	• Explain various architecture and components of the Embedded System
CO2: Apply	• Develop Arduino program for various interface
CO3: Apply	• Select the Arduino shield for a given application
CO4: Design	• Implement various communication interface for data exchange between Arduino and other devices/systems
CO5: Design, Document, Project Management (mini project)	• Develop Arduino based system for a given real-life application in a team

3. Course Duration: The course duration is of **40 sessions of 60 minutes each.**

4. Course Contents:

Unit No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	• **Introduction to Embedded System:** Difference between microprocessor and microcontroller, RISC and CISC architecture, Harward and Von Neumann Architecture, Applications of Embedded System	2	4
II	• **The Arduino Family:** Types of Arduino Devices, Software Compatible Devices, Use of Arduino Devices, Arduino Technical Features	3	4
III	• **Programming Arduino:** Installing Arduini IDE, Writing Arduino programs (Sketches), Write programs to implement various mathematical operation, Install various Arduino Libraries (EEPROM, Ethernet, GSM, LCDs, SD Card, Firmata, Servo, SPI, SoftwareSerail, TFT, WiFI, Wire, Esplora, USB, Keyboard, Mouse etc.)	6	10
IV	• **Arduino Shields:** Electrical and Physical Characteristics of the Shields, Various types of Ardunio Shields	2	6
V	• **Serial Communication** Serial hardware and software, Serial message protocol, sending data from Arduino to Computer, Sending formatted text and numerical data from Arduino,	5	10

	Receiving serial data in Arduino, Sending and receiving multiple text in single message, Sending and receiving binary values, sending serial data to two devices, receiving serial data from two devices, Sending serial data to file in a computer		
VI	• **Interfacing with Input Devices and Sensor** Using serial communication, Digital and Analog Inputs, Interfacing Sensors (Movement, Motion, Light, Temperature, Humidity, distance, sound, vibration, mouse, GPS, acceleration, gyroscope, RFID Tag)	6	12
VII	• **Interfacing Output Devices** Visual output (LEDs, Seven Segment, LED Matrix), Physical output (Servo, Solenoid, Stepper motor, Brushless DC motor), Remote Controlling External Devices using IR remote controller, Using Displays (Text LCD display, Graphical LCD display)	6	08
VIII	• **Using Times and Dates** Creating delays, determine duration, measuring duration of pulse, Arduino as a Clock, create alarm, using real-time clock	4	08
IX	• **Various Communications** Communication using I2C, SPI; Wireless communication, Ethernet and Networking,	6	08

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive classroom discussions
- Application Demonstration
- Mini Project

6. Suggestive list of experiments

1. Interface digital input and output devices to the Arduino
2. Interface ADC with Arduino
3. Interface DAC with Arduino
4. Interface Seven Segment LED display with Arduino
5. Interface LED Matrix display with Arduino
6. Interface LCD Display with Arduino
7. Interface Graphical LCD with Arduino
8. Interface Temperature and Humidity Sensors with Arduino
9. Interface Real Time Clock with Arduino
10. Interface I2C device with Arduino
11. Interface SPI device with Arduino
12. Interface and control speed of servo motor with Arduino
13. Interface and control speed of stepper motor with Arduino
14. Design IR remote control system using Arduino to operate remote devices
15. Interface Arduino through wireless communication to other devices
16. Interface Arduino through ethernet

7. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

8. Software Tools:

1. **Arduino IDE (https://www.arduino.cc/en/software)**
2. **Arduino Simulator – UnoArduSim (https://www.sites.google.com/site/unoardusim/services)**

9. Reference Books:

No.	Author	Name of the Book	Publisher
1.	Michael Margolis	Arduino Cookbook	O'Reilly
2.	J. M. Hughes	Arduino: A Technical Reference	O'Reilly
3.	Ashwin Pajankar	Arduino Made Simple	BPB
4.	Muhammad Ali Mazidi, Sarmad Naimi, Sepher Naimi	The AVR Microcontroller and Embedded System: Using Assembly and C	Pearson

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Internet of Things
Subject Code: 639409

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:-

Course Outcome Component	Course Outcome (Learner will be able to)
CO1	• Understand the concept of Internet of Things (IoT)
CO2	• Understand and compare various sensors and will be able to select sensors for IoT application
CO3	• Design IoT applications in different domain and be able to analyze their performance
CO4	• Identify real life problem and suggest solution using IoT
CO5	• Understand security issues in IoT Application

3. Course Duration: The course duration is of **40 sessions of 60 minutes each.**

4. Course Contents:

Unit No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	• **Introduction to Internet of Things:** Application areas of IoT, Characteristics of IoT, Things in IoT, IoT stack, Enabling technologies, IoT challenges, IoT levels, IoT and cyber physical system, IoT & WSN.	06	10
II	• **Sensors, Microcontrollers, and Their Interfacing:** Characteristics of Sensor, Sensor interfacing, Types of sensors, Controlling sensors, Actuators, Types of Actuators, Microcontrollers, ARM.	06	10
III	• **Protocols for IoT& Connectivity Technology:** Messaging protocols, Transport protocols, IPv4, IPv6, URI, MQTT, CoAP, XMPP, AMQP. IEEE 802.15.4, Zigbee, 6LoWPAN, RFID, NFC, Bluetooth.	08	15
IV	• **Application Building with IoT:** Various application of IoT: Food, Healthcare, Lavatory maintenance, Water quality, Warehouse, Retail, Driver Assistance, Collision impact.	06	10
V	• **Arduino and Raspberry Pi:** **Arduino:** Architecture, Component, IDE, Programme Element, Function Library, Random Number, Interrupts. **Raspberry Pi:** Architecture, Compatible Peripherals, Add-Ons, and Accessories, PIN Configuration, Case Study, Programming and Implementation of IoT with Raspberry Pi.	08	15

VI	• **Security and Privacy in IoT:** Various security issues and need, Security architecture, Network & Transport Layer Challenges, Authorization Mechanism, Security Framework for IoT, Privacy in IoT Networks.	06	10

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Text Books:

No.	Author	Name of the Book	Publisher
1	Vasudevan, Nagrajan and Sundaram	Internet of Things	WileyIndia
2	Dr. Jeeva Jose	Internet of Things	Khanna Book Publishing
3	Rajkumar Buyya, Amir Vahid, Dastjerdi	Internet of Things Principles and Paradigm	ELSEVIER

8. Reference Books:

No.	Author	Name of the Book	Publisher
1	David Hence at el,	IoT Fundamentals	CiscoPress
2	Yashavant Kanetkar, Shrirang Korde	21 IoT Experiments,	BPB
3	Adrian McEwen, Hakim Cassimally	Designing the Internet of Things	Wiley
4	Raj Kamal	Internet of Things Architecture and Design Principles	McGraw Hill

9. Sample Practical List

Students are expected to

1. Understand architecture of Arduino and Raspberry PI.
2. Understand various sensors and integration of sensors with Arduino/Raspberry PI.
3. Design an application like Smart Home using interfacing of various sensors and Arduino/Raspberry PI module.

List of suggestive practical list is as follows.

1. Introduction to various sensors and various actuators & its Application. Example are as under
    a) PIR Motion Sensor.
    b) Rain Drop Sensor.
    c) Moisture Sensor.
    d) Temperature Sensor.
    e) Touch Sensor.
    f) Infrared Sensor.
    g) Servo Motor.
    h) RFID Sensor.
    i) Bluetooth Module.
    j) Wi-Fi Module.

2. Experiment using Arduino Uno to measure the distance of any object using Ultrasonic Sensor.
3. Create a circuit using Arduino and sensors. Perform experiment using Arduino to Learn Working of Servo Motor.
4. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
5. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
6. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
7. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when '1'/'0' is received from smartphone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thingspeak cloud.
10. To install MySQL database on Raspberry Pi and perform basic SQL queries.
11. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.
12. Write a program on Arduino/Raspberry Pi to subscribe to MQTT broker for temperature data and print it.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 3rd Semester
Subject Name: Network & Cyber Security
Subject Code: 639410

With effective from academic year 2020-21

1. Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA (M)	ESE (V)	PA (I)	
3	-	2	4	70	30	30	20	150

2. Course Outcomes:

Course Outcome Component	Course Outcome (Learner will be able to)
CO1: Concepts of Network Security	• Learn and describe about various network security and cyber security concepts, devices used to enhance security of networks.
CO2: Concepts of Cryptography and IP Security	• Learn and describe about various cryptographic techniques, digital signatures, various hashing algorithms and their importance and internet protocol architecture.
CO3:Network Scanning and Identification	• Learn and identify various devices present across network, identify the open ports on the active devices, identify the OS information and banner information of various servers and machines.
CO4:Network Monitoring and Analysis	• Learn and capture traffic from the active network, analyse packets & protocols and create their own Network Monitoring System
CO5: Wireless Security	• Learn and describe various security authentications and standards used, detect and mitigate various attacks performed on wireless network infrastructure.

3. Course Duration: The course duration is of **40 sessions of 60 minutes each.**

4. Course Contents:

Module No:	Contents	No. of Sessions	70 Marks (External Evaluation)
I	**Introduction to Network Security & Cyber Security Concepts:** Network Security and its need, CIA (Confidentiality, Integrity, Availability), AAA (Authentication, Authorization, Accounting), Network Devices (Host, Router, Switch, Bridge, etc..) on Each Layer of OSI Model, Working of DNS, DHCP, IDS (Intrusion Detection System), IPS (Intrusion Prevention System), Firewall and its types, Web Proxies, Internet Security Protocols	6	15
II	**Introduction to Cryptography & IP Security:** Key Terms: Encryption, Decryption, Plain Text, Cipher Text, Secret Code, Types of Cryptographic Functions, Secret Key Cryptography, Public Key Cryptography, Hashing, Hash	5	15

	Algorithms, Digital Signatures, IP Security Architecture – Authentication Header, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange (IKE).		
III	**Network Security Assessment - 1:** Passive Information Gathering: IP Address & Domain Identification, Banner Grabbing (Netcat, Telnet & other tools), Identifying Domain Ownership (WHOIS, DNS Lookup & other tools), Active Information Gathering: Detecting Active Systems, ICMP Ping, Port Scanning and its techniques, Port Scanning Tools (Nmap, Zenmap, Superscan & other tools), OS Fingerprinting – Active & Passive.	11	15
IV	**Network Security Assessment - 2:** Physical Interception, Traffic Capturing Tools (Wireshark), Packet Analysis, Protocol Analysis, Traffic Timeline Analysis, Setting up your own Network Intrusion Detection System (SNORT-NIDS).	10	15
V	**Wireless Security:** Wi-Fi basics – Wireless Clients and NICs, Wireless Access Points (WAP), Wireless Communication Standards, Wi-Fi Security – 802.1x Authentication, Wireless LAN Threats – Wardriving (NetStumbler, Kismet), Eavesdropping, Rogue and Unauthorized Access Points, Evil Twin Attack, DOS, WLAN Encrytion Flaws: Cracking WPA/WPA2 PSK, Decrypting WEP and WPA Packets, ARP poising and MAC spoofing, Security Wireless Security.	8	10

5. Pedagogy:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

6. Evaluation:

Students shall be evaluated on the following components:

A	**Internal Evaluation**	**(Total - 20 Marks)**
	• Continuous Evaluation Component	10 marks
	• Class Presence & Participation	10 marks
B	**Mid-Semester examination**	**(30 Marks)**
C	**End –Semester Examination(Theory)**	**(70 Marks)**
D	**End –Semester Examination(Practical/Viva)**	**(30 Marks)**

7. Reference Books:

No.	Author	Name of the Book	Publisher
1.	Michael Gregg	Build Your Own Security Lab: A Field Guide for Network Testing	Wiley Publishing
2.	Charlie Kaufman,	Network Security: Private Communication	Pearson Indian

	Radia Perlman and Mike Speciner	in a Public World	Education Services Ltd.
3	William Stallings	Cryptography and Network Security: Principles and Practice	Pearson
4.	Lisa Bock	Learn Wireshark	Packt Publishing
5.	Nicholas Marsh	Nmap® Cookbook: The Fat-Free Guide to Network Scanning	Create space Independent Pub
6.	ED Wilson	Networking Monitoring And Analysis : A Protocol Approach to Troubleshooting	Prentice Hall PTR
7.	Chris McNab	Network Security Assessment: Know your Network	O'Reilly

8. Practical

List of suggestive practical list is as follows.

Sr. No.	Suggested practical List
1	To study various wired & wireless network devices based on layers of OSI Model.
2	To study various attacks based on layers of OSI Model.
3	To perform following operations using Netcat: • Port Scanning & Port Listening • Banner Grabbing • File/Data Transfer • Chat Server
4	Study the use of network reconnaissance tools like dig, traceroute, nslookup to gather information about networks and domain registrars.
5	To perform Open Source Intelligence (OSINT) about any specific domain. (WHOIS, DNS Lookup & other Tools) – A Passive Information Gathering Technique.
6	To perform port scanning using various methods & techniques provided by Nmap or Zenmap.
7	To implement a packet capturing tool (Wireshark) and capture the real time traffic.
8	To study & analyse the captured packets for different protocols & search queries using Wireshark.
9	Observe performance in promiscuous as well as non-promiscous mode in Wireshark and also show that packets can be traced based on different filters.
10	Use the Nessus tool to scan the network for vulnerabilities.
11	To implement/configure Intrusion Detection System for Log Collection based on default & customized rules. (Ex. Snort IDS)
12	To study the features of firewall in providing Network Security and policy implementation on any basic firewall.
13	To implement whitelisting & blacklisting policy in the firewall.
14	To study ARP Protocol & perform ARP poising attack.
15	To study WEP, WPA2PSK and perform WEP, WPA cracking.
16	To study and report on latest Network Security Crimes, Network Security Challenges and Solutions to overcome them.
17	To Implement Caesar cipher encryption-decryption & Playfair cipher encryption-decryption
18	To write a program to generate SHA-1 hash & to implement a digital signature algorithm.
19	To perform various encryption-decryption techniques with cryptool. Technique like caesar cipher, Monoaplhabetic cipher, polyalphabetic cipher, rectangular cipher, columnar cipher, Hill cipher etc.

GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 4th Semester
Subject Name: Software Project - 3
Subject Code: 649401

With effective from academic year 2021-22

1. Learning Objectives:

- To solve industrial (or society or research) problems.
- To plan, schedule, and monitor the software project
- Development, coding, and testing of a large project cohesively.
- Documentation of project

2. Prerequisites:

Software Engineering, Programming / Coding language, RDBMS

3. Guidelines for Project

- It is recommended that the team should be about 2-3 students.
- The project should be free from plagiarism of any kind.
- Internal guides (i.e. the regular faculty members) must be allocated to projects.
- Project plan along with the division of work amongst teammates would have been prepared and got approved within a maximum of 5 days of the start of the project.
- Coding standards should be followed meticulously. At the minimum, the code should be self documented, modular, and should use the meaningful naming convention.
- It is advisable that object-oriented methodology is used with the reusability of classes and code, etc..
- The output reports must include MIS reports, if applicable.
- The documentation should include a chapter on "Learning during Project Work", i.e. "Experience of Journey during Project Duration".
- It is strongly recommended that Data structure/Database design is included in the report. At least portions of code (preferably full code) are mandatory. The student may be asked to write the code related to the project during the examination.
- If a student is compelled to follow certain instructions (by the external, i.e. organization's Guide) which he/she does not agree to, such a student must prepare a supplementary report to document his/her version and present it to the examiners if such a need arises.
- Internal guides (i.e. the regular faculty members) must devote the time allocated as per the timetable to guide the students for the project. The time allocation will be in accordance with the scheme for the 6th semester project as given.
- Internal guides should preferably visit external guide to track the project
- Project document should be printed on both sides of paper.

4. Accomplishments of the student after completing the course:

- Doing the project will enable the student to go through rich experience in developing large projects. Such an experience will include encountering various technical issues, finding sources to resolve the issues and finally finding the solution of all these issues satisfactorily.
- Thinking analytically, analyzing, and synthesizing requirements and complicated information for getting a good comprehension of the solution methodology to be adopted.
- Ability to document and write well.
- Organizing the time effectively.
- Working with teammates and generating substantial output of the efforts.
- It will prepare the students for analyzing and programming for industrial problem and large projects work in future

5. Documentation:

- The project has to be well-documented in the form of a Project Report (at least 50 pages comprising of the design, data dictionary, source code, screenshots, etc.).
- Format: Print out on both the side of page with single line spacing. Use Times New Roman of size 10 for normal text.

TABLE OF CONTENTS

1. Introduction
    - 1.1. Existing System
    - 1.2. Need for the New System
    - 1.3. Objective of the New System
    - 1.4. Problem Definition
    - 1.5. Core Components
    - 1.6. Project Profile
    - 1.7. Assumptions and Constraints
    - 1.8. Advantages and Limitations of the Proposed System
2. Requirement Determination & Analysis
    - 2.1. Requirement Determination
    - 2.2. Targeted Users
3. System Design
    - 3.1. Use Case Diagram
    - 3.2. Class Diagram
    - 3.3. Interaction Diagram
    - 3.4. Activity Diagram
    - 3.5. Data Dictionary
4. Development
    - 4.1. Coding Standards
    - 4.2. Screen Shots
5. Agile Documentation
    - 5.1 Agile Project Charter
    - 5.2 Agile Roadmap / Schedule
    - 5.3 Agile Project Plan
    - 5.4 Agile User Story (Minimum 3 Tasks)
    - 5.5 Agile Release Plan
    - 5.6 Agile Sprint Backlog
    - 5.7 Agile Test Plan
    - 5.8 Earned-value and burn charts
6. Proposed Enhancements
7. Conclusion
8. Bibliography

6. Evaluation Parameters :

- Evaluation of the projects would be done considering the framework available at the Institute. The main parameter of assessment would be the ability of the students to code.
- Though the project and domain specific knowledge would be assessed for, the evaluation would predominantly depend on the students' ability to explain, modify or revise of code.
- Coding standards should have been implemented.

- Though the project would be evaluated for the entire team, the examiner should emphasize on the contribution of each team member in the project development
- Total Marks (700 = 500 External + 200 Internal)

Break up of External Marks (500)

Documentation (more specifically, Correctness and completeness of UML diagrams, and relationship between Class diagram & Database structure)	100
Explanation of Analysis & Design	100
Explanation of Code (To test the ability to explain how to code few functionalities used in the project)	200
Presentation	100