**Janardan Bhagat Shikshan Prasarak Sanstha's**

# CHANGU KANA THAKUR
# ARTS, COMMERCE & SCIENCE COLLEGE,
# NEW PANVEL (AUTONOMOUS)

**Re-accredited 'A+' Grade by NAAC**
**'College with Potential for Excellence' Status Awarded by UGC**
**'Best College Award' by University of Mumbai**

**Program: B.Sc**

**Revised Syllabus of S.Y.B.Sc. Information Technology**
**Choice Based Credit & Grading System (60:40)**
**w.e.f. Academic Year 2019-2020**

## Preamble of the Syllabus:

The S.Y.B.Sc. Information Technology programme is started with an aim to make the learners employable and impart industry oriented training.

The main objectives of the course are:

- To develop skills to identify and apply suitable data structure for the given real world problem.
- To understand and apply python scripting language and implement it as a successful developer in IT industry.
- To obtain a theoretical understanding of data communication and computer networks, and gaining practical experience in installation, monitoring, and troubleshooting of current LAN systems.
- To learn why Java is useful for the design of desktop and web applications etc.
- To impart detailed knowledge of techniques for the analysis and design of complex software intensive systems.
- To develop modern day technologies like animation, interactive multimedia, or construction graphics, which can lead computer graphics majors in any number of directions and also develop Web technologies, produce spectacular CD-ROMs, or become a specialist in computer-aided design (CAD)?

## Objectives of the Course:

The new syllabus is aimed to achieve the objectives. The syllabus spanning two semesters covers the industry relevant courses. The students will be ready for the jobs available in different fields like:

- Python Programming
- Data Structures
- Computer Networks
- Advanced SQL
- Applied Mathematics
- Core Java
- Introduction to Embedded systems
- Computer Oriented Statistical Tech
- Software Engineering
- Computer Graphics and Animation.

## Course Outcome: By the end of the course, a student should develop the Ability:

- Learners are able to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS.
- Learners are able to apply knowledge of computing and mathematical skills appropriate to the discipline.
- Learners are able to describe those aspects of mobile programming that make it unique from programming for other platforms.
- Learners are able to understand hardware and software design requirements of embedded systems and hardware and software design requirements of embedded systems.

## Scheme of Examination

**A) Internal Assessment: 40 %** **40 Marks**

| **Sr. No.** | **Particular** | | **Marks** |
|---|---|---|---|
| 01 | One periodical class test / online examination to be conducted in the given semester | | 20 Marks |
| 02 | One case study/ project with presentation based on curriculum to be assessed by the teacher concerned | | 15 Marks |
| | Presentation | 10 Marks | |
| | Written Document | 05 Marks | |
| 03 | Active participation in routine class instructional deliveries and overall conduct as a responsible learner, mannerism and articulation and exhibit of leadership qualities in organizing related academic activities | | 05 Marks |

**Question Paper Pattern for Class Test**

| **Question No.** | **Particular** | **Marks** |
|---|---|---|
| Q-1 | Match the Column / Fill in the Blanks / Multiple Choice Questions/ Answer in One or Two Lines (Concept based Questions) ( 1 Marks / 2 Marks each) | 10 Marks |
| Q-2 | Answer in Brief (Attempt any Two of the Three)<br>(5 Marks each) | 10 Marks |

B) **Semester End Examination: 60 %** **60 Marks**

- Duration: The examination shall be of 2 hours duration.

**Theory question paper pattern**

1. There shall be five questions each of 12 marks.
2. All questions shall be compulsory with internal options.
3. Question may be subdivided into sub-questions a, b, c… and the allocation of marks depends on the weightage of the unit.

➢ **Passing Standard**

The learners to pass a course shall have to obtain a minimum of 40% marks in aggregate for each course where the course consists of Internal Assessment and Semester End Examination. The learners shall obtain minimum of 40% marks (i.e. 16 out of 40) in the Internal Assessment and 40% marks in Semester End Examination (i.e. 24 Out of 60) separately, to pass the course and minimum of Grade D, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Internal Assessment and Semester End

Examination together.

## Choice Based Credit Grading and Semester System (CBCGS)
## S.Y.B. Sc. Information Technology Syllabus
## To be implemented from the Academic year 2019-2020
## SEMESTER III

| **Course Code** | **Course Type** | **Course Title** | **Credits** |
|---|---|---|---|
| UIT3PYP | Skill Enhancement Course | Python Programming | 2 |
| UIT3DST | Core Subject | Data Structures | 2 |
| UIT3C-N | Core Subject | Computer Networks | 2 |
| UIT3ADS | Core Subject | Advanced SQL | 2 |
| UIT3MAT | Core Subject | Applied Mathematics | 2 |
| UIT3PPP | Skill Enhancement Course Practical | Python Programming Practical | 2 |
| UIT3DSP | Core Subject Practical | Data Structures Practical | 2 |
| UIT3CNP | Core Subject Practical | Computer Networks Practical | 2 |
| UIT3 ASP | Core Subject Practical | Advanced SQL Practical | 2 |
| UIT3MPP | Core Subject Practical | Mobile Programming Practical | 2 |
| | Total Credits | | **20** |

## Choice Based Credit Grading and Semester System (CBCGS)
## S.Y.B. Sc. Information Technology Syllabus
## To be implemented from the Academic year 2019-2020
## SEMESTER IV

| **Course Code** | **Course Type** | **Course Title** | **Credits** |
|---|---|---|---|
| UIT4CJ V | Skill Enhancement Course | Core Java | 2 |
| UIT4EMB | Core Subject | Introduction to Embedded ystems | 2 |
| UIT4COS | Core Subject | Computer Oriented Statistical Techniques | 2 |
| UIT4SWE | Core Subject | Software Engineering | 2 |
| UIT4CGA | Core Subject | Computer Graphics and Animation | 2 |
| UIT4CJ P | Skill Enhancement Course Practical | Core Java Practical | 2 |
| UIT4ESP | Core Subject Practical | Introduction to Embedded Systems Practical | 2 |
| UIT4COP | Core Subject Practical | Computer Oriented Statistical Techniques Practical | 2 |
| UIT4SEP | Core Subject Practical | Software Engineering Practical | 2 |
| UIT4CGP | Core Subject Practical | Computer Graphics and Animation Practical | 2 |
| | **Total Credits** | | **20** |

# Semester III

| **Course Code**<br>UIT3PYP | **Python Programming** |
|---|---|

**Objectives**
The objective of this course is to introduce the concept of the basic programming language with C++.

**Expected Learning Outcomes:**
1) Learners should be able to understand how C++ improves C with object-oriented features.
2) Learners should be able to learn how to write inline functions for efficiency and performance
3) Learners should be able to write programs that are very efficient in memory usage.

| | | |
|---|---|---|
| **I** | **Introduction:** The Python Programming Language, History, features, Installing Python, Running Python program, Debugging : Syntax Errors, Runtime Errors, Semantic Errors, Experimental Debugging, Formal and Natural Languages, The Difference Between Brackets, Braces, and Parentheses,<br>**Variables and Expressions** Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Interactive Mode and Script Mode, Order of Operations.<br>**Conditional Statements:** if, if-else, nested if –else<br>**Looping**: for, while, nested loops<br>**Control statements:** Terminating loops, skipping specific conditions | **12** |
| **II** | **Functions:** Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Stack Diagrams, Fruitful Functions and Void Functions, Why Functions? Importing with from, Return Values, Incremental Development, Composition, Boolean Functions, More Recursion, Leap of Faith, Checking Types<br>**Strings:** A String Is a Sequence, Traversal with a for Loop, String Slices, Strings Are Immutable, Searching, Looping and Counting, String Methods, The in Operator, String Comparison, String Operations. | **12** |
| **III** | **Lists:** Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods<br>**Tuples and Dictionaries:** Tuples, Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in Tuple Functions<br>Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods<br>**Files:** Text Files, The File Object Attributes, Directories<br>**Exceptions:** Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions | **12** |
| **IV** | **Regular Expressions** – Concept of regular expression, various types of regular expressions, using match function.<br>**Classes and Objects:** Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding<br>**Multithreaded Programming:** Thread Module, creating a thread, synchronizing threads, | **12** |

| | | |
|---|---|---|
| | multithreaded priority queue<br>**Modules:** Importing module, Creating and exploring modules, Math module, Random module, Time module | |
| **V** | **Creating the GUI Form and Adding Widgets:**<br>**Widgets:** Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessagebox.<br>Handling Standard attributes and Properties of Widgets.<br>**Layout Management:** Designing GUI applications with proper Layout Management features.<br>**Look and Feel Customization**: Enhancing Look and Feel of GUI using different appearances of widgets.<br>**Storing Data in Our MySQL Database via Our GUI:** Connecting to a MySQL database from Python, Configuring the MySQL connection, Designing the Python GUI database, Using the INSERT command, Using the UPDATE command, Using the DELETE command, Storing and retrieving data from MySQL database. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| UIT3PPP | 1. **Write the program for the following:**<br>a. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.<br>b. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.<br>c. Write a program to generate the Fibonacci series.<br>d. Write a function that reverses the user defined value.<br>e. Write a function to check the input value is Armstrong and also write the function for Palindrome.<br>f. Write a recursive function to print the factorial for a given number.<br>2. **Write the program for the following:**<br>a. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.<br>b. Define a function that computes the *length* of a given list or string.<br>c. Define a procedurehistogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:<br>****<br>*********<br>*******<br>3. **Write the program for the following:**<br>a. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.<br>b. Take a list, say for example this one:<br>a=[1,1,2,3,5,8,13,21,34,55,89]<br>and write a program that prints out all the elements of the list that are less than 5. |

4. **Write the program for the following:**

a. Write a program that takes two lists and returns True if they have at least one common member.
b. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.
c. Write a Python program to clone or copy a list

5. **Write the program for the following:**

a. Write a Python script to sort (ascending and descending) a dictionary by value.

b. Write a Python script to concatenate following dictionaries to create a new one.
Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
c. Write a Python program to sum all the items in a dictionary.

6. **Write the program for the following:**

a. Write a Python program to read an entire text file.
b. Write a Python program to append text to a file and display the text.
c. Write a Python program to read last n lines of a file.

7. **Write the program for the following:**

a. Design a class that store the information of student and display the same
b. Implement the concept of inheritance using python

8. **Write the program for the following:**

a. Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it.
Now open a new file and save it in the same directory. You should now be able to import your own module like this: importgeometry
b. Write a program to implement exception handling.

9. **Write the program for the following:**

a. Try to configure the widget with various options like: bg=”red”, family=”times”, size=18

b. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.

10. **Design the database applications for the following:**

| | |
|---|---|
| | a. Design a simple database application that stores the records and retrieve the same.<br>b. Design a database application to search the specified record from the database.<br>c. Design a database application to that allows the user to add, delete and modify the records. |

**Reference Books:**

1) "Think Python" , Allen Downey, First edition, O'Reilly publication
2) "An Introduction to Computer Science using Python 3" , Jason Montojo, First Edition, SPD publication.
3) "Python GUI Programming Cookbook", Burkhard A. Meier, Packt Edition
4) "Introduction to Problem Solving with Python", E Balagurusamy , First Edition ,Tata McGraw Hill.
5) "Murach's Python Programming" , Joel Murach & Michael Urban, First Edition, SPD Publication
6) "Object Oriented Programming in Python",Michael H. Glodwasser,First Edition, Pearson Prentice Hall Publication
7) "Exploring Python", Budd, First Edition, TMH Publication

| **Course Code** UIT3DST | **Data Structures** |
|---|---|

**Objectives**

The objective of this course is to introduce the basic knowledge of algorithms and analysis procedure and determine the complexity of given algorithms and techniques.

**Expected Learning Outcomes**

1) Learners should be able to understand and remember algorithms and its analysis procedure.
2) Learners should be able to understand and examine the concept of data structures through ADT including List, Stack, Queue.
3) Learners should be able to compute the complexity of various algorithms.
4) Learners should be able to understand appropriate sorting/searching technique for given problem.
5) Learners should be able to design advance data structure using Non-linear data structure like tree and graph.

| **Unit** | **Details** | **lectures** |
|---|---|---|
| **I** | **Introduction:** Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Data structure vs. File Organization, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation.<br>**Array**: Introduction, One Dimensional Array, Memory Representation of One | **12** |

| | | |
|---|---|---|
| | Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multi-Dimensional Arrays, Sparse Arrays, Sparse Matrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays. | |
| **II** | **Linked List:** Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and De-allocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Other List, Merging Two Linked Lists, Splitting a List into Two Lists, Reversing One way linked List, Circular Linked List, Applications of Circular Linked List, Two way Linked List, Traversing a Two way Linked List, Searching in a Two way linked List, Insertion of an element in Two way Linked List, Deleting a node from Two way Linked List, Header Linked List, Applications of the Linked list, Representation of Polynomials, Storage of Sparse Arrays, Implementing other Data Structures. | **12** |
| **III** | **Stack**: Introduction, Operations on the Stack Memory Representation of Stack, Array Representation of Stack, Applications of Stack, Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion**.**<br>**Queue:** Introduction, Queue, Operations on the Queue, Memory Representation of Queue, Array representation of queue, Linked List Representation of Queue, Circular Queue, Some special kinds of queues, Deque, Priority Queue, Application of Priority Queue, Applications of Queues. | **12** |
| **IV** | **Sorting and Searching Techniques**<br>Bubble, Selection, Insertion, Merge Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search.<br>**Tree:** Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree, Heap, Memory Representation of Heap, Operation on Heap, Heap Sort.<br>**Advanced Tree Structures:** Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, 2-3 Tree, B-Tree. | **12** |
| **V** | **Hashing Techniques**<br>Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing<br>**Graph**: Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| UIT3DSP | **1. Implement the following:**<br>a. Write a program to store the elements in 1-D array and perform the operations like searching, sorting and reversing the elements. [Menu Driven]<br>b. Read the two arrays from the user and merge them and display the elements in sorted order.[Menu Driven]<br>c. Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven]<br>**2. Implement the following for Linked List:**<br>a. Write a program to create a single linked list and display the node elements in reverse order. |

| | |
|---|---|
| | b. Write a program to search the elements in the linked list and display the same<br>c. Write a program to create double linked list and sort the elements in the linked list.<br>**3.Implement the following for Stack:**<br>a. Write a program to implement the concept of Stack with Push, Pop, Display and Exit operations.<br>b. Write a program to convert an infix expression to postfix and prefix conversion.<br>c. Write a program to implement Tower of Hanoi problem.<br>**4. Implement the following for Queue:**<br>a. Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.<br>b. Write a program to implement the concept of Circular Queue<br>c. Write a program to implement the concept of Deque.<br>**5. Implement the following sorting techniques:**<br>a. Write a program to implement bubble sort.<br>b. Write a program to implement selection sort.<br>c. Write a program to implement insertion sort.<br>**6. Implement the following data structure techniques:**<br>a. Write a program to implement merge sort.<br>b. Write a program to search the element using sequential search.<br>c. Write a program to search the element using binary search.<br>**7. Implement the following data structure techniques:**<br>a. Write a program to create the tree and display the elements.<br>b. Write a program to construct the binary tree.<br>c. Write a program for inorder, postorder and preorder traversal of tree.<br>**8. Implement the following data structure techniques:**<br>a. Write a program to insert the element into maximum heap.<br>b. Write a program to insert the element into minimum heap.<br>**9. Implement the following data structure techniques**:<br>a. Write a program to implement the collision technique.<br>b. Write a program to implement the concept of linear probing.<br>**10. Implement the following data structure techniques**:<br>a. Write a program to generate the adjacency matrix.<br>b. Write a program for shortest path diagram. |

**Reference Books:**

1. A Simplified Approach to Data Structures - Lalit Goyal, Vishal Goyal, Pawan Kumar
2. An Introduction to Data Structure with Applications - Jean – Paul Tremblay and Paul Sorenson.
3. Data Structure and Algorithm -Maria Rukadikar  Tata McGraw Hill

| **Course Code**<br>**UIT3C-N** | **Computer Networks** |
|---|---|

**Objectives:**

On completion of this course, a learner will be able to understand about computer network organization and implementation, obtaining a theoretical understanding of data communication and computer networks, and gaining practical experience in installation, monitoring, and troubleshooting of current LAN systems.

**Expected Learning Outcomes:**

1) Learners will be able to describe the general principles of data communication.

| | | |
|---|---|---|
| 2) Learners will be able to describe how computer networks are organized with the concept of layered Approach.<br>3) Learners will be able to describe how signals are used to transfer data between nodes.<br>4) Learners will be able to learn about different network protocols.<br>5) Learners will be able to identify different network technologies | | |
| **I** | **Introduction:** Data communications, networks, network types, Internet history, standards and administration.<br>**Network Models:** Protocol layering, TCP/IP protocol suite, The OSI model.<br>**Introduction to Physical layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance.<br>**Digital and Analog transmission:** Digital-to-digital conversion, analog-to-digital conversion, transmission modes, digital-to-analog conversion, analog-to-analog conversion. | **12** |
| **III** | **Data Link Control:** DLC services, data link layer protocols, HDLC, Point-to-point protocol.<br>**Media Access Control**: Random access, controlled access, channelization, Wired LANs – Ethernet Protocol, standard ethernet, fast ethernet, gigabit ethernet, 10 gigabit ethernet,<br>**Wireless LANs:** Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks.<br>**Connecting devices and Virtual LANs.** | **12** |
| **IV** | **Introduction to the Network Layer**: Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets, Internet Protocol, ICMPv4, Mobile IP<br>**Unicast Routing:** Introduction, routing algorithms, unicast routing protocols.<br>**Next generation IP:** IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6. | **12** |
| **V** | **Introduction to the Transport Layer**: Introduction, Transport layer protocols (Simple protocol, Stop-and-wait protocol, Go-Back-n protocol, Selective repeat protocol, Bidirectional protocols), Transport layer services, User datagram protocol, Transmission control protocol,<br>**Standard Client0Server Protocols:** World wide-web and HTTP, FTP, Electronic mail, Telnet, Secured Shell, Domain name system. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| **UIT3CNP** | **1. IPv4 Addressing and Subnetting**<br>A .Given an IP address and network mask, determine other information about the IP address such as:<br>• Network address<br>• Network broadcast address<br>• Total number of host bits<br>• Number of hosts<br>B .Given an IP address and network mask, determine other information about the IP address such as:<br>• The subnet address of this subnet<br>• The broadcast address of this subnet<br>• The range of host addresses for this subnet<br>• The maximum number of subnets for this subnet mask<br>• The number of hosts for each subnet |

| | |
|---|---|
| | • The number of subnet bits<br>• The number of this subnet<br>**2.** Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.<br>**3.** Configure IP static routing..<br>**4.** Configure IP routing using RIP.<br>**5.** Configuring Simple OSPF.<br>**6.** Configuring DHCP server and client.<br>**7.** Create virtual PC based network using virtualization software and virtual NIC.<br>**8.** Configuring DNS Server and client.<br>**9.** Configuring OSPF with multiple areas.<br>**10**.Use of Wireshark to scan and check the packet information of following protocols<br>• HTTP<br>• ICMP<br>• TCP<br>• SMTP<br>• POP3 |

**Reference Books:**

a. "Data Communication and Networking", Behrouz A. Forouzan , Tata McGraw Hill , Fifth Edition , 2013.

b. "TCP/IP protocol suite", Behrouz A. Forouzan , Tata McGraw Hill , Fifth Edition , 2013.

c. "Computer Networks", Andrew Tanenbaum , Pearson , Fifth Edition,2013

<table>
<tr><td><b>Course Code<br>UIT3ADS</b></td><td colspan="2"><b>Advanced SQL</b></td></tr>
<tr><td colspan="3"><b>Objectives</b><br>The objective of this course is to introduce the concept of the <b>Advanced SQL</b> with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases<br><br><b>Expected Learning Outcomes</b><br>1) Learners should be able to design the database schema with the use of appropriate data types for Storage of data in database.<br>2) Learners should be able to create, manipulate, query and back up the databases.<br>3) Learners should be able to design relational databases.<br>4) Learners should be able to perform basic of PL/SQL programming.<br>5) Learners should be able to perform the advanced PL/SQL Programming.</td></tr>
<tr><td><b>I</b></td><td>Structured Query Language: Writing Basic SQL Select Statements, Restricting and Sorting Data, Single-Row Functions, Aggregating Data using Group Functions, Manipulating Data, Creating and Managing Tables</td><td><b>12</b></td></tr>
</table>

| | | |
|---|---|---|
| **II** | **Advanced SQL:** Subqueries, Creating Views, Creating Other Database Objects(Sequences, Indexes and Synonyms) Controlling User Access, Using SET operators, Date Time Functions, Joins (Displaying Data from Multiple Tables),<br>**Constraints :**<br>Constraints, types of constrains, Integrity constraints<br>WITH Clause, Hierarchical retrieval | **12** |
| **III** | **PL-SQL:** Beginning with PL / SQL, Identifiers and Keywords, Operators, Expressions, Sequences, Overview and benefits of PL/SQL, Subprograms, types of PL/SQL blocks, Simple Anonymous Block, Identifiers, types of Identifiers, Declarative Section, variables, Scalar Data Types, The %TYPE Attribute, Executable Statements, PL/SQL Block Syntax, Comment the Code, Convert Data Types, Nested Blocks, Operators. Invoke SELECT Statements in PL/SQL, Save and Discard Transactions | **12** |
| **IV** | **Control Structures**: Conditional processing using IF Statements and CASE Statements, Loop Statement, While Loop Statement, For Loop Statement, the Continue Statement<br>**Explicit Cursors**: Declare the Cursor, Open the Cursor, Fetch data from the Cursor, Close the Cursor, Cursor FOR loop, The %NOTFOUND and %ROWCOUNT Attributes<br>**Exception Handling**: Handle Exceptions with PL/SQL, Propagate Exceptions<br>Composite Type: PL/SQL Records, The %ROWTYPE Attribute, INDEX BY Tables, INDEX BY Table Methods | **12** |
| **V** | **Stored Procedures**: Create, Call, and Remove Stored Procedures, Implement Procedures Parameters and Parameters Modes<br>**Stored Functions** Create, Call, and Remove a Stored Function, advantages of using Stored Functions, the steps to create a stored function, Invoke User-Defined Functions in SQL Statements<br>**Packages:** advantages of Packages, components of a Package, Develop a Package, enable visibility of a Package's Components, Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer,<br>**Triggers:** the Trigger Event Types and Body, Create DML Triggers using the CREATE TRIGGER Statement, Identify the Trigger Event Types, Body, and Firing (Timing), Statement Level Triggers and Row Level Triggers, Manage, Test and Remove Triggers. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| **UIT3DMP** | 1. **Practical 1: Select queries**<br>a. Select queries on single table using alias, where and order by clause.<br>b. Select queries on single table using aggregate<br>c.<br>2. **Practical 2: Select queries using joins and unions**<br>a. Querying data from multiple tables using all types of joins.<br>b. Querying data from multiple tables using all types of joins.<br>3. **Practical 3: Subqueries, DML and DDL**<br>a. Querying single and multiple tables using subqueries. |

| | |
|---|---|
| | b. Manipulating data (Insert, update and delete)<br>c. Creating simple tables and tables with constraints.<br>**4. Practical 4: Creating database objects, using set operators**<br>a. Creating Views, Sequences, Indexes and synonyms.<br>b. Using set operators, date-time functions,<br>**5. Practical 5: Working with advanced subqueries and WITH clause**<br>a. Multiple column subqueries, subqueries in from clause,<br>b. WITH Clause and hierarchical retrieval.<br>**6. Practical 6: Basic PL/SQL, INDEX BY tables, PL/SQL Record and FOR loop.**<br>a. Creating anonymous PL/SQL blocks.<br>b. Define, create, and use INDEX BY tables and a PL/SQL record.<br>**7. Practical 7: Cursors, Exceptions and procedures issuing DML and query commands.**<br>a. Cursors with parameters to process a number of rows from multiple tables.<br>b. Create exception handlers for specific situations.<br>**8. Practical 8: Functions and Stored Procedures**<br>a. Creating and invoking functions from SQL statements.<br>b. Creating and invoking stored procedures.<br>**9. Practical 9: Working with packages**<br>a. Create package specifications and package bodies. Invoke the constructs in the packages.<br>b. Create a package containing an overloaded function.<br>c.<br>**9. Practical 9: Working with Large Objects and triggers**<br>a. Create statement triggers.<br>b. Create row triggers.<br>**10. Practical 10: Working with INSTEAD OF triggers**<br>a. Create instead of triggers for views. |

**Reference Books:**

1) Murach's Oracle SQL and PLSQL by Joel Murach, Murach and Associates.
2) Oracle database 11g : hands on SQL/PL SQL by Satish Asnani (PHI) EEE edition
3) Programming with PL/SQL for Beginners, H. Dand, R. Patil and T. Sambare, First Edition X- Team,2011
4) PL/SQL Programming, Ivan Bayross , First Edition, BPB 2010

| **Course Code**<br>**UIT3MAT** | **Applied Mathematics** |
|---|---|
| **Objectives**<br>The learners will understand the concepts of applications of the methods for solving different mathematical structures. This course introduces the advance learning of matrices and complex | |

numbers, differential equations, Laplace transforms and the error functions.

**Expected Learning Outcomes**

1) Learners should be able to solve the matrix using different methods and solve the hyperbolic functions of the complex numbers.
2) Learners should be able to identify the origin and applications of differential equations, solve initial value problems and linear DE with constant coefficient.
3) Learners should be able to understand laplace transform and apply inverse laplace to find solution of differential equations.
4) Learners should be able to find area and volume using integrals.
5) Learners should be able to find the error functions and understand the properties of beta gamma functions.

| | | |
|---|---|---|
| **I** | **Matrices:** Inverse of a matrix, Properties of matrices, Elementary Transformation, Rank of Matrix, Echelon or Normal Matrix, Inverse of matrix, Linear equations, Linear dependence and linear independence of vectors, Linear transformation, Characteristics roots and characteristics vectors, Properties of characteristic vectors, Caley-Hamilton Theorem, Similarity of matrices, Reduction of matrix to a diagonal matrix which has elements as characteristics values.<br>**Complex Numbers:** Complex number, Equality of complex numbers, Graphical representation of complex number(Argand's Diagram), Polar form of complex numbers, Polar form of x+iy for different signs of x,y, Exponential form of complex numbers, Mathematical operation with complex numbers and their representation on Argand's Diagram, Circular functions of complex angles, Definition of hyperbolic function, Relations between circular and hyperbolic functions, Inverse hyperbolic functions, Differentiation and Integration, Graphs of the hyperbolic functions, Logarithms of complex quality, j(=i)as an operator(Electrical circuits) | **12** |
| **II** | **Equation of the first order and of the first degree:** Separation of variables, Equations homogeneous in x and y, Non-homogeneous linear equations, Exact differential Equation, Integrating Factor, Linear Equation and equation reducible to this form, Method of substitution.<br>**Differential equation of the first order of a degree higher than the first:** Introduction, Solvable for p (or the method of factors), Solve for y, Solve for x, Clairaut's form of the equation, Methods of Substitution, Method of Substitution.<br>**Linear Differential Equations with Constant Coefficients:** Introduction, The Differential Operator, Linear Differential Equation f(D) y = 0, Different cases depending on the nature of the root of the equation f(D) = 0, Linear differential equation f(D) y = X, The complimentary Function, The inverse operator 1/f(D) and the symbolic expiration for the particular integral 1/f(D) X; the general methods, Particular integral : Short methods, Particular integral : Other methods, Differential equations reducible to the linear differential equations with constant coefficients | **12** |
| **III** | **The Laplace Transform:** Introduction, Definition of the Laplace Transform, Table of Elementary Laplace Transforms, Theorems on Important Properties of Laplace Transformation, First Shifting Theorem, Second Shifting Theorem, The Convolution Theorem, Laplace Transform of an Integral, Laplace Transform of Derivatives,<br>**Inverse Laplace Transform:** Shifting Theorem, Partial fraction Methods, Use of Convolution Theorem, Solution of Ordinary Linear Differential Equations with Constant Coefficients, Solution of Simultaneous Ordinary Differential Equations, | **12** |

| | | |
|---|---|---|
| | Laplace Transformation of Special Function, Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function(Unit Impulse Function) | |
| **IV** | **Multiple Integrals:** Double Integral, Change of the order of the integration, Double integral in polar co-ordinates, Triple integrals.<br>**Applications of integration:** Areas, Volumes of solids. | **12** |
| **V** | **Beta and Gamma Functions –** Definitions**,** Properties and Problems. Duplication formula.<br>**Differentiation Under the Integral Sign**<br>**Error Functions** | **12** |

**Books and References:**

1) A text book of Applied Mathematics Vol I, P. N. Wartikar and J. N. Wartikar ,PuneVidyathi Graha
2) Applied Mathematics II , P. N. Wartikar and J. N. Wartikar ,Pune Vidyathi Graha
3) Higher Engineering Mathematics, Dr. B.S.Grewal, Khanna publications.

| **Course Code** | **Practical List** |
|---|---|
| **UIT3MPP** | **List of Practical**<br>Setting up CORDOVA, PhoneGAP Project and environment.<br>1. • Creating and building simple "Hello World" App using Cordova<br>• Adding and Using Buttons<br>• Adding and Using Event Listeners<br>2. • Creating and Using Functions<br>• Using Events<br>• Handling and Using Back Button<br>3. • Installing and Using Plugins<br>• Installing and Using Battery Plugin<br>• Installing and Using Camera Plugin<br>4. • Installing and Using Contacts Plugin<br>• Installing and Using Device Plugin<br>• Installing and Using Accelerometer Plugin<br>5. • Install and Using Device Orientation plugin<br>• Install and Using Device Orientation plugin<br>• Create and Using Prompt Function<br>6. • Installing and Using File Plugin<br>• Installing and Using File Transfer Plugin<br>• Using Download and Upload functions<br>7. • Installing and Using Globalization Plugin<br>• Installing and Using Media Plugin |

| | |
|---|---|
| | • Installing and Using Media Capture Plugin<br>8. • Installing and Using Network Information Plugin<br>• Installing and Using Splash Screen Plugin<br>• Installing and Using Vibration Plugin |

**Reference Books:**

1. Apache Cordova 4 Programming John M. Wargo 1st Addison-Wesley Professional
2. Apache Cordova in ActionRaymond Camden 1st Manning Publications

## Semester IV

| **Course Code**<br>UIT4CJV | **Core Java** |
|---|---|

**Objectives**

The objective of this course is to introduce the concept of the java programming language and understand its fundamentals.

**Expected Learning Outcomes:**

1) Students will be introduced with the basic concepts of java programming.
2) Students will be able to implement java programs using control structures, iteration.
3) Students will use the advance class features including inheritance, polymorphism and overloading, overriding, interfacing, abstract classes and develop efficient and reusable codes.
4) Learners will be made familiar with multithreading, IO File handling and exception handling techniques.
5) Students will be able to design, develop and execute AWT application.

| | | |
|---|---|---|
| **I** | **Introduction:** History, architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, java platform, java development kit, Lambda Expressions, Methods References, Type Annotations, Method Parameter Reflection, setting the path environment variable, Java Compiler And Interpreter, java programs, java applications, main(), public, static, void, string[] args, statements, white space, case sensitivity, identifiers, keywords, comments, braces and code blocks, variables, variable name<br>**Data types:** primitive data types, Object Reference Types, Strings, Auto boxing, operators and properties of operators, Arithmetic operators, assignment operators, increment and decrement operator, relational operator, logical operator, bitwise operator, conditional operator. | **12** |
| **II** | **Control Flow Statements**: The If…Else If…Else Statement, The Switch…Case Statement<br>**Iterations:** The While Loop, The Do … While Loop, The For Loop, The Foreach Loop, Labelled Statements, The Break And Continue Statements, The Return Statement<br>**Classes:** Types of Classes, Scope Rules, Access Modifier, Instantiating Objects From A Class, Initializing The Class Object And Its Attributes, Class Methods, Accessing A Method, Method Returning A Value, Method's Arguments, Method Overloading, Variable Arguments [Varargs], Constructors, this Instance, super Instance, Characteristics Of Members Of A Class, constants, this instance, static fields of a class, static methods of a class, garbage collection | **12** |

| | | |
|---|---|---|
| **III** | **Inheritance:** Derived Class Objects, Inheritance and Access Control, Default Base Class Constructors, this and super keywords.<br>Abstract Classes And Interfaces, Abstract Classes, Abstract Methods, Interfaces, What Is An Interface? How Is An Interface Different From An Abstract Class?, Multiple Inheritance, Default Implementation, Adding New Functionality, Method Implementation, Classes V/s Interfaces, Defining An Interface, Implementing Interfaces.<br>**Packages:** Creating Packages, Default Package, Importing Packages, Using A Package. | **12** |
| **IV** | **Enumerations, Arrays:** Two Dimensional Arrays, Multi-Dimensional Arrays, Vectors, Adding Elements To A Vector, Accessing Vector Elements, Searching For Elements In A Vector, Working With The Size of The Vector.<br>**Multithreading:** the thread control methods, thread life cycle, the main thread, creating a thread, extending the thread class.<br>**Exceptions:** Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause<br>**Byte streams:** reading console input, writing console output, reading file, writing file, writing binary data, reading binary data, getting started with character streams, writing file, reading file | **12** |
| **V** | **Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes and inner classes.<br>**Abstract Window Toolkit:** Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas. Components – Labels, Buttons, Check Boxes, Radio Buttons, Choice Menus, Text Fields, Text, Scrolling List, Scrollbars, Panels, Frames<br>**Layouts:** Flow Layout, Grid Layout, Border Layout, Card Layout. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| UIT4CJP | **1.Java Basics**<br>a. Write a Java program that takes a number as input and prints its multiplication table upto 10.<br>b. Write a Java program to display the patterns.<br>c. Write a Java program to print the area and perimeter of a circle.<br>**2.Use of Operators**<br>a. Write a Java program to add two binary numbers.<br>b. Write a Java program to convert a decimal number to binary number and vice versa.<br>c. Write a Java program to reverse a string.<br>**3.Java Data Types**<br>a. Write a Java program to count the letters, spaces, numbers and other characters of an input string.<br>b. Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value.<br>c. Find the smallest and largest element from the array<br>**4.Methods and Constructors**<br>a. Designed a class SortData that contains the method asec() and desc().<br>b. Designed a class that demonstrates the use of constructor and destructor.<br>c. Write a java program to demonstrate the implementation of abstract class.<br>**5.Inheritance**<br>a. Write a java program to implement single level inheritance.<br>b. Write a java program to implement method overriding<br>c. Write a java program to implement multiple inheritance.<br>**6.Packages and Arrays** |

| | |
|---|---|
| | a. Create a package, Add the necessary classes and import the package in java class.<br>b. Write a java program to add two matrices and print the resultant matrix.<br>c. Write a java program for multiplying two matrices and print the product for the same.<br>**7.Vectors and Multithreading**<br>a. Write a java program to implement the vectors.<br>b. Write a java program to implement thread life cycle.<br>c. Write a java program to implement multithreading.<br>**8.File Handling**<br>a. Write a java program to open a file and display the contents in the console window.<br>b. Write a java program to copy the contents from one file to other file.<br>c. Write a java program to read the student data from user and store it in the file.<br>**9.GUI and Exception Handling**<br>a. Design a AWT program to print the factorial for an input value.<br>b. Design an AWT program to perform various string operations like reverse string, string concatenation etc.<br>c. Write a java program to implement exception handling.<br>**10.GUI Programming.**<br>a. Design an AWT application that contains the interface to add student information and display the same.<br>b. Design a calculator based on AWT application.<br>c. Design an AWT application to generate result marks sheet. |

**Reference Books:**

1) "Core Java 8 for Beginners" , Vaishali Shah, Sharnam Shah, First edition, SPD publication
2) "Java: The Complete Reference" , Herbert Schildt, Ninth Edition, McGraw Hill.

| **Course Code**<br>UIT4EMB | **Introduction to Embedded System** |
|---|---|

**Objectives**

To introduce learner with the core components of embedded system and 8051 programing in C.
They will also be able to differentiate  types of operating system.

**Expected Learning Outcomes:**

1) Become familiar with classification, characteristics, core components of embedded system
2) Become familiar with memory, types of memory, registers
3) Acquire skills in 8051 programing in C
4) Acquire skills for selecting microcontroller and developing basic applications
5) Become familiar with different types of operating system and its characteristics.

| | | |
|---|---|---|
| **I** | **Introduction:** Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems<br>**Core of embedded systems:** microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little endian processors, Application specific ICs, Programmable logic devices, COTS, sensors and actuators, communication interface, embedded firmware, other system components.<br>**Characteristics and quality attributes of embedded systems:** Characteristics, operational and non-operational quality attributes. | **12** |

| | | |
|---|---|---|
| **II** | **Embedded Systems – Application and Domain Specific:** Application specific – washing machine, domain specific - automotive.<br>**Embedded Hardware:** Memory map, i/o map, interrupt map, processor family, external peripherals, memory – RAM , ROM, types of RAM and ROM, memory testing, CRC ,Flash memory.<br>**Peripherals:** Control and Status Registers, Device Driver, Timer Driver - Watchdog Timer | **12** |
| **III** | **The 8051 Microcontrollers:** Microcontrollers and Embedded processors, Overview of 8051 family.8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory.<br>**8051 Programming in C:**<br>Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs. | **12** |
| **IV** | **Designing Embedded System with 8051 Microcontroller:** Factors to be considered in selecting a controller, why 8051 Microcontroller, Designing with 8051.<br>**Programming embedded systems:** structure of embedded program, infinite loop, compiling, linking and debugging | **12** |
| **V** | **Real Time Operating System (RTOS):** Operating system basics, types of operating systems, Real-Time Characteristics, Selection Process of an RTOS.<br>**Design and Development:** Embedded system development Environment – IDE, types of file generated on cross compilation, disassembler/ de-compiler, simulator, emulator and debugging, embedded product development life-cycle, trends in embedded industry. | **12** |

| **Course Code** | **Practical List** |
|---|---|
| UIT4ESP | **1)Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects**.<br>a. Programming<br>b. Execution<br>c. Debugging<br><br>2A) Configure timer control registers of 8051 and develop a program to generate given time delay.<br>2B) To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them.<br><br>3A) Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED's. Simulate binary counter (8 bit) on LED's<br><br>3B) To interface 8 LEDs at Input-output port and create different patterns.<br>3C) To demonstrate timer working in timer mode and blink LED without using any loop delay routine.<br><br>4A) Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen. Signify end of message by carriage return.<br><br>4B) To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay. |

| | |
|---|---|
| | 4C) Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope.<br>5A) Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope.<br>5B) Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051.<br>6) Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction.<br>7) Generate traffic signal.<br>8) Implement Temperature controller.<br>9) Implement Elevator control.<br>10) Using Flash Magic<br>a. To demonstrate the procedure for flash programming for reprogrammable embedded system board using Flash Magic<br>b. To demonstrate the procedure and connections for multiple controllers programming of same type of controller with same source code in one go, using flash magic. |

**Reference Books:**

1) Introduction to embedded systems ,Shibu K V, First, Tata Mcgraw-Hill
2) Embedded Systems Rajkamal Tata Mcgraw-Hill

| **Course Code**<br>UIT4COS | **Computer Oriented Statistical Techniques** |
|---|---|

**Objectives**

The objective of this course is to provide an understanding for the learners on statistical concepts to include measures of dispersion probability distribution, sampling estimation, hypothesis testing, regression and correlation analysis.

**Expected Learning Outcomes:**

By completing this course the learners will be able to perform:

1. To calculate and apply measures of dispersion.
2. To apply discrete and continuous probability distribution to various problems.
3. The test of hypothesis as well as calculate confidence interval for a population parameter and learn the concept to p-value.
4. Learn non parametric test such as the Chi- Square test for Independence as well as goodness of fit.
5. to compute and interpret the results of bivariate and multivariate regression and correlation analysis and to perform ANOVA. Be able to perform multiple regression using computer software R.

| | | |
|---|---|---|
| **I** | | **12** |

| | | |
|---|---|---|
| | **The Mean, Median, Mode, and Other Measures of Central Tendency**: Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean , The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean ,The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode, The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency.<br>**The Standard Deviation and Other Measures of Dispersion**: Dispersion, or Variation, The Range, The Mean Deviation, The Semi-Interquartile Range, The 10–90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties of the Standard Deviation, Charlie’s Check, Sheppard’s Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient of Variation, Standardized Variable; Standard Scores, Software and Measures of Dispersion.<br>**Introduction to R:** Basic syntax, data types, variables, operators, control statements, R-functions, R –Vectors, R – lists, R Arrays . | |
| **II** | **Moments, Skewness, and Kurtosis** : Moments , Moments for Grouped Data ,Relations Between Moments , Computation of Moments for Grouped Data, Charlie’s Check and Sheppard’s Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness, and Kurtosis, Software Computation of Skewness and Kurtosis.<br>**Elementary Probability Theory**: Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Probability Distributions, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinatorial Analysis, Combinations, Stirling’s Approximation to n!, Relation of Probability to Point Set Theory, Euler or Venn Diagrams and Probability.<br>**Elementary Sampling Theory** : Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement,<br>Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory . | **12** |
| **III** | **Statistical Estimation Theory:** Estimation of Parameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error.<br>**Statistical Decision Theory:** Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating-Characteristic Curves; the Power of a Test, p-Values for Hypotheses Tests, Control Charts, Tests Involving Sample Differences, Tests Involving Binomial Distributions.<br>**Statistics in R:** Mean, Median, Mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R | **12** |
| **IV** | **Small Sampling Theory**: Small Samples, Student’s t Distribution, Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma , Degrees of Freedom, The F Distribution.<br>**The Chi-Square Test**: Observed and Theoretical Frequencies, Definition of chi-square, Significance Tests, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates’ Correction for Continuity, Simple Formulas for Computing chi-square, Coefficient of Contingency, Correlation of Attributes, Additive Property of chi-square | 12 |

| | | |
|---|---|---|
| **V** | **Curve Fitting and the Method of Least Squares:** Relationship Between Variables, Curve Fitting, Equations of Approximating Curves, Freehand Method of Curve Fitting, The Straight Line, The Method of Least Squares, The Least-Squares Line, Nonlinear Relationships, The Least-Squares Parabola, Regression, Applications to Time Series, Problems Involving More Than Two Variables.<br>**Correlation Theory:** Correlation and Regression, Linear Correlation, Measures of Correlation, The Least-Squares Regression Lines, Standard Error of Estimate, Explained and Unexplained Variation, Coefficient of Correlation, Remarks Concerning the Correlation Coefficient, Product-Moment Formula for the Linear Correlation Coefficient, Short Computational Formulas, Regression Lines and the Linear Correlation Coefficient, Correlation of Time Series, Correlation of Attributes, Sampling Theory of Correlation, Sampling Theory of Regression | **12** |

| **Course Code** | **Practical List** |
|---|---|
| UIT4COP | 1.Using R execute the basic commands, array, list and frames.<br>2.Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operations.<br>3.Using R Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram<br>4.Using R import the data from Excel / .CSV file and Perform the above functions.<br>5.Using R import the data from Excel / .CSV file and Calculate the standard deviation, variance, co-variance.<br>6. Using R import the data from Excel / .CSV file and draw the skewness.<br>7. Import the data from Excel / .CSV and perform the hypothetical testing.<br>8. Import the data from Excel / .CSV and perform the Chi-squared Test.<br>9.Using R perform the binomial and normal distribution on the data.<br>10.Perform the Linear Regression using R.<br>11.Compute the Least squares means using R.<br>12.Compute the Linear Least Square Regression |

**Reference Books:**

1. STATISTICS, Murray R Spiegel, Larry J. Stephens, McGRAW –HILL ITERNATIONAL, Fourth edition.
2. FUNDAMENTAL OF MATHEMATICAL STATISTICS S.C. GUPTA and V.K. KAPOOR, SULTAN CHAND and SONS, ELEVENTH EDITION.
3. A Practical Approach using R , R.B. Patil, H.J. Dand and R. Bhavsar , SPD publication, First edition.

| **Course Code**<br>UIT4SWE | **Software Engineering** |
|---|---|

**Objectives**

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.
3. Ability to develop, maintain and evaluate large-scale software systems.
4. To produce efficient, reliable, robust and cost-effective software solutions.

**Expected Learning Outcomes:**

1. Ability to perform independent research and analysis.
2. To communicate and coordinate competently by listening, speaking, reading and writing English for technical and general purposes.

| | | |
|---|---|---|
| 3. Ability to work as an effective member or leader of software engineering teams.<br>4. To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.<br>5. Ability to understand and meet ethical standards and legal responsibilities. | | |
| **I** | **Introduction:** What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing, Maintenance etc.<br>Software Requirements: Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements.<br>**Software Processes:**<br>Process and Project, Component Software Processes.<br>**Software Development Process Models.**<br>• Waterfall Model.<br>• Prototyping.<br>• Iterative Development.<br>• Rational Unified Process.<br>• The RAD Model<br>• Time boxing Model.<br>**Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods. | **12** |
| **II** | **Socio-technical system:** Essential characteristics of socio technical systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computers, Dealing Legacy Systems.<br>**Critical system:** Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems.<br>**Requirements Engineering Processes:** Feasibility study, Requirements elicitation and analysis, Requirements Validations, Requirements Management.<br>**System Models:** Models and its types, Context Models, Behavioural Models, Data Models, Object Models, Structured Methods. | **12** |
| **III** | **Architectural Design:** Architectural Design Decisions, System Organisation, Modular Decomposition Styles, Control Styles, Reference Architectures.<br>**User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation.<br>**Project Management:** Software Project Management, Management activities, Project Planning, Project Scheduling, and Risk Management.<br>**Quality Management:** Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics. | **12** |
| **IV** | **Verification and Validation:** Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods. **Software Testing:** System Testing, Component Testing, Test Case Design, Test Automation.<br>**Software Measurement**: Size-Oriented Metrics, Function-Oriented Metrics, Extended Function Point Metrics<br>**Software Cost Estimation:** Software Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing | **12** |
| **V** | **Process Improvement:** Process and product quality, Process Classification, Process Measurement, Process Analysis and Modelling, Process Change, The CMMI Process Improvement Framework.<br>**Service Oriented Software Engineering:** Services as reusable components,<br>Service Engineering, Software Development with Services.<br>**Software reuse:** The reuse landscape, Application frameworks, Software product lines, | **12** |

| | | |
|---|---|---|
| | COTS product reuse.<br>**Distributed software engineering**: Distributed systems issues, Client–server computing, Architectural patterns for distributed systems, Software as a service. | |

| **Course Code** | **Practical List** |
|---|---|
| UIT4SEP | 1. Study and implementation of class diagrams.<br>2. Study and implementation of Use Case Diagrams.<br>3. Study and implementation of Entity Relationship Diagrams.<br>4. Study and implementation of Sequence Diagrams.<br>5. Study and implementation of State Transition Diagrams.<br>6. Study and implementation of Data Flow Diagrams.<br>7. Study and implementation of Collaboration Diagrams.<br>8. Study and implementation of Activity Diagrams.<br>9. Study and implementation of Component Diagrams.<br>10. Study and implementation of Deployment Diagrams. |

**Reference Books:**

1. Software Engineering, edition, Ian Somerville Pearson Education. Edition Ninth
2. Software Engineering Pankaj Jalote Narosa Publication
3. Software engineering, a practitioner's approach Roger Pressman Tata Mcgraw-hill Seventh edition
4. Software Engineering principles and practice WS Jawadekar Tata Mcgraw-hill

| **Course Code**<br>UIT4CGA | **Computer Graphics and Animation** |
|---|---|

**Objectives**

To make the learners aware of the different algorithms that are actually used for developing different types of animations. This subject aims in making the learners capable of designing different animations programmatically.

**Expected Learning Outcomes:**

1. Learners will know the different algorithms used for computer graphics.
2. Learners will gain knowledge about projections and transformations.
3. Acquire basic knowledge about 3D effects
4. Learners will know different types of surfaces and their basic algorithms.
5. Understand Animation and storage processing.

| | | |
|---|---|---|
| **I** | **Introduction to Computer Graphics:**<br>Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays.<br>**Scan conversion** – Digital Differential Analyzer (DDA) algorithm, Bresenhams' Line drawing algorithm. Bresenhams' method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms– Cyrus-Beck, | **12** |

| | | |
|---|---|---|
| | Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components. • Time boxing Model.<br>**Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods. | |
| **II** | **Two-Dimensional Transformations:**<br>Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations.<br>**Three-Dimensional Transformations:**<br>Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections. | **12** |
| **III** | **Viewing in 3D**<br>Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid.<br>**Light:** Radiometry, Transport, Equation, Photometry<br>**Color:** Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance | **12** |
| **IV** | **Visible-Surface Determination:**<br>Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods.<br>**Plane Curves and Surfaces:**<br>Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, Representation of Space Curves, Cubic Splines, , Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces. | **12** |
| **V** | **Computer Animation:**<br>Principles of Animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects.<br>**Image Manipulation and Storage:**<br>What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering. | **12** |

| **Course Code** | **Practical List** |
|---|---|

| | |
|---|---|
| UIT4CGP | **1. Solve the following:**<br>a. Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them. Draw a co-ordinate axis at the center of the screen.<br>**2. Solve the following**:<br>a. Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each regionwith appropriate message.<br>b. Draw a simple hut on the screen.<br>**3. Draw the following basic shapes in the center of the screen :**<br>i. Circle ii. Rectangle iii. Square iv. Concentric Circles v. Ellipse vi. Line<br>**4. Solve the following:**<br>a. Develop the program for DDA Line drawing algorithm.<br>b. Develop the program for Bresenham's Line drawing algorithm.<br>**5. Solve the following:**<br>a. Develop the program for the mid-point circle drawing algorithm.<br>b. Develop the program for the mid-point ellipse drawing algorithm.<br>**6. Solve the following:**<br>a. Write a program to implement 2D scaling.<br>b. Write a program to perform 2D translation<br>**7. Solve the following:**<br>a. Perform 2D Rotation on a given object.<br>b. Program to create a house like figure and perform the following operations.<br>i. Scaling about the origin followed by translation.<br>ii. Scaling with reference to an arbitrary point.<br>iii. Reflect about the line y = mx + c.<br>**8. Solve the following:**<br>a.Write a program to implement Cohen-Sutherland clipping.<br>b. Write a program to implement Liang - Barsky Line Clipping Algorithm<br>**9. Solve the following:**<br>a. Write a program to fill a circle using Flood Fill Algorithm.<br>b. Write a program to fill a circle using Boundary Fill Algorithm.<br>**10. Solve the following:**<br>a.Develop a simple text screen saver using graphics functions.<br>b.Perform smiling face animation using graphic functions.<br>c.Draw the moving car on the screen. |

**Reference Books:**

1. Computer Graphics - Principles and Practice J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes Pearson 2nd edition.
2. Computer Graphics Hearn, Baker Pearson 2nd edition.
3. Fundamentals of Computer Graphics Steve Marschner, Peter Shirley 4th edition.