

AC- 02 /03 /19

Item No- 2.31



**RayatShikshanSanstha's
KARMAVEER BHAAURAO PATIL COLLEGE, VASHI.
NAVI MUMBAI
(AUTONOMOUS COLLEGE)
Sector-15- A, Vashi, Navi Mumbai - 400 703**

Syllabus for S.Y.B.Sc.Information Technology

Program: B.Sc. Information Technology

Course: S.Y.B.Sc. Information Technology

**(Choice Based Credit, Grading and Semester System
with effect from the academic year 2019-2020)**

RayatShikshanSanstha's

KarmaveerBhauraoPatil College Vashi, Navi Mumbai

Autonomous College

[University of Mumbai]

Syllabus for Approval

| Sr. No | Heading | Particulars |
|---------------|---|---|
| 1 | Title of Course | S.Y.B.Sc. Information Technology |
| 2 | Eligibility for Admission | 12th Maths |
| 3 | Passing Marks | 40% |
| 4 | Ordinances/Regulations (if any) | |
| 5 | No. of Years/Semesters | One year/Two semester |
| 6 | Level | U.G. |
| 7 | Pattern | Semester |
| 8 | Status | Revised |
| 9 | To be implemented from Academic year | 2019-2020 |

Preamble of the Syllabus:

The B.Sc. Information Technology programme was started in 2001 with an aim to make the students employable and impart industry oriented training. The main objectives of the course are:

- To think analytically, creatively and critically in developing robust, extensible and highly maintainable technological solutions to simple and complex problems.
- To apply their knowledge and skills to be employed and excel in IT professional careers and/or to continue their education in IT and/or related post graduate programmes.
- To be capable of managing complex IT projects with consideration of the human, financial and environmental factors.
- To work effectively as a part of a team to achieve a common stated goal.
- To adhere to the highest standards of ethics, including relevant industry and organizational codes of conduct.
- To communicate effectively with a range of audiences both technical and non-technical.
- To develop an aptitude to engage in continuing professional development. The new syllabus is aimed to achieve the objectives. The syllabus spanning three years covers the industry relevant courses. The students will be ready for the jobs available in different fields like:
 - Software Development (Programming)
 - Website Development
 - Mobile app development
 - Embedded Systems Development & Programming
 - Software Testing
 - Networking
 - Database Administration
 - System Administration
 - Cyber Law Consultant
 - GIS (Geographic Information Systems)
 - IT Service Desk
 - Security

And many others

The students will also be trained in communication skills and green computing

Syllabus for S.Y.B.Sc. Information Technology

Objectives of the Program:

- To acquaint students with the fundamental of computer hardware and software in information technology
- To develop analytical skills and critical thinking through application of theory knowledge into practical course
- To construct and apply knowledge of programming, and appreciate the relationship between several programming languages and other disciplines
- To enable students to understand IT and its industrial and social context

Program Outcome:

By the end of the course, a student should develop the ability:

- Student will understand, coherently and effectively about various basic components of computers.
- Student can improve their computer literacy, their basic understanding of operative systems and a working knowledge of software commonly used in academic and professional environments.
- Student can able to develop basic skills in practical of Information Technology and its industrial applications.
- Student can do Academic and Professional Presentations - Designing and delivering an effective presentation and developing the various IT skills to the electronic databases.
- Student can develop ability to solve IT-oriented security issues and protocols
- Student can definitely design and implement a web page.
- Student can improve communication and business management skills, especially in providing technical support.

Scheme of examination for Each Semester:

Continuous Internal Evaluation: 40 Marks(Common Test-20 Marks & 20 Marks for- Assignment, Projects, Group discussion, Open book test, online test etc.)

Semester End Examination: 60 Marks will be as follows -

| | | |
|----------------|---|---|
| I. | Theory: The Semester End Examination for theory course work will be conducted as per the following scheme. | |
| | Each theory paper shall be of two and half hour duration. | |
| | All questions are compulsory and will have internal options. | |
| | Q – I | From Unit – I (having internal options.) 10 M |
| | Q – II | From Unit – II (having internal options.) 10 M |
| | Q – III | From Unit – III (having internal options.) 10 M |
| | Q – IV | From Unit – IV (having internal options.) 10 M |
| | Q-V | From Unit – V (having internal options.)10 M |
| Q-VI | From Unit – VI (having internal options.) 10 M | |
| II. | Practical | The Semester End Examination for practical course work will be conducted as per the following scheme. |
| Sr. No. | Particulars of Semester End Practical Examination | Marks% |
| 1 | Laboratory Work | 40 |
| 2 | Journal | 05 |
| 3 | Viva | 05 |

| | | |
|--|--------------|-----------|
| | TOTAL | 50 |
|--|--------------|-----------|

Semester – III

| Course Code | Course Title | Credits | Course Code | Course Title | Credits | Total Credits | Total |
|--------------------|---|----------------|--------------------|---------------------------------------|----------------|----------------------|--------------|
| UGIT301 | Core Java | 02 | UGIT3P1 | Core Java Practical | 02 | 04 | 20 |
| UGIT302 | Applied Mathematics | 02 | UGIT3P2 | Applied Mathematics Practical | 02 | 04 | |
| UGIT303 | Linux System Administration | 02 | UGIT3P3 | Linux System Administration Practical | 02 | 04 | |
| UGIT304 | Database Management System | 02 | UGIT3P4 | Database Management System Practical | 02 | 04 | |
| UGIT305 | Elective (Any one) 1. Data Structures | 02 | UGIT3P5 | 1. Data Structure Practical | 02 | 04 | |
| UGIT306 | 2. System Software | | UGIT3P6 | 2. System Software Practical | | | |

Semester – IV

| Course Code | Course Title | Credits | Course Code | Course Title | Credits | Total Credits | Total |
|--------------------|--|----------------|--------------------|--|----------------|----------------------|--------------|
| UGIT401 | Computer Graphics and Animation | 02 | UGIT4P1 | Computer Graphics and Animation Practical | 02 | 04 | 20 |
| UGIT402 | Introduction to Embedded Systems | 02 | UGIT4P2 | Introduction to Embedded Systems Practical | 02 | 04 | |
| UGIT403 | Computer Oriented Statistical Techniques | 02 | UGIT4P3 | Computer Oriented Statistical Techniques Practical | 02 | 04 | |
| UGIT404 | Software Project Management | 02 | UGIT4P4 | Software Project Management Practical | 02 | 04 | |

| | | | | | | | |
|---------|--|-----------|---------|---|-----------|-----------|--|
| UGIT405 | Elective (Any one) 1. Python Programming. 2. Data warehousing | 02 | UGIT4P5 | Elective (Any one) 1. Python Programming Practical 2. Data warehousing Practical | 02 | 04 | |
| UGIT406 | | | UGIT4P6 | | | | |

Semester – III

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Course Name: Core Java | | Course Code: UGIT301 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives:

The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the pre-requisites of Core java.

Expected Learning Outcomes:

- Student will be able to understand the object oriented programming concepts using Java.
- Student will be able to implement the control flow statements, iteration and classes.
- Student will be able to understand, design, implement and evaluate classes.
- Student will be able to implement inheritance and packages.
- Student will be able to understand and implement multithreading.
- Student will be able to implement event handling and AWT packages.
- Student will be able to understand, design, and implement applets and JDBC.

| Unit | Details | Lectures |
|-------------|--|-----------------|
| I | <p>Introduction: Architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, java development kit, 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</p> <p>Data types and Operators: Primitive data types, Object Reference Types, Strings, Auto boxing, Operators and properties of operators, Arithmetic operators, assignment operators, increment</p> | 10 |

| | | |
|-----|---|----|
| | <p>and decrement operator, relational operator, logical operator, bitwise operator, conditional operator.</p> <p>String Handling: String class and StringBuffer Class, Formatting string data using format() method.</p> <p>Arrays: Two Dimensional Arrays, Multi-Dimensional Array.</p> | |
| II | <p>Control flow Statements: The If...Else If...Else Statement, The Switch...Case Statement</p> <p>Iterations: The While Loop, The Do ... While Loop, The For Loop, The Foreach Loop, Labeled Statements, The Break And Continue Statements, The Return</p> <p>Classes: Statement 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], Abstract Classes, Abstract Methods, 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.</p> | 10 |
| III | <p>Inheritance: Derived Class Objects, Inheritance and Access Control, Default Base Class Constructors, this and super keywords. 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.</p> <p>Packages:Creating Packages, Default Package, Importing Packages, Using A Package.</p> | 10 |
| IV | <p>Multithreading:The thread control methods, thread life cycle, the main thread, creating a thread, extending the thread class.</p> <p>Exceptions:Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause.</p> <p>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</p> | 10 |
| V | <p>Event Handling: Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes and inner classes.</p> | 10 |

| | | |
|----|---|----|
| | <p>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.</p> <p>Layouts: Flow Layout, Grid Layout, Border Layout, Card Layout.</p> | |
| VI | <p>Applet: Applet Life Cycle, appletviewer tool, Applet HTML Tags, Passing parameters to Applet, repaint() and update() method.</p> <p>JDBC: Design of JDBC, JDBC configuration, Executing SQL statement, Query Execution, Scrollable and updatable result sets, row sets, metadata, Transaction.</p> | 10 |

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|---|-----------------------------|-------------|---------|------|
| 1. | Core Java 8 for Beginners | Vaishali Shah, Sharnam Shah | SPD | 1st | 2015 |
| 2. | Java: The Complete Reference | Herbert Schildt | McGraw Hill | 9th | 2014 |
| 3. | Murach's beginning Java with Net Beans | Joel Murach , Michael Urban | SPD | 1st | 2016 |
| 4. | Core Java, Volume I: Fundamentals | Hortsman | Pearson | 9th | 2013 |
| 5. | Core Java, Volume II: Advanced Features | Gary Cornell and Hortsman | Pearson | 8th | 2008 |
| 6. | Core Java: An Integrated Approach | R. NageswaraRao | DreamTech | 1st | 2008 |

| B. Sc (Information Technology) | | Semester – III | |
|--|------------------------------|-----------------------------|--------------|
| Course Name: Core Java Practical | | Course Code: UGIT3P1 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

List of Practicals:

1. Java Basics

- Write a Java program that takes a number as input and prints its multiplication table upto 10
- Write a Java program to display the following pattern.

**
*

c. Write a Java program to print the area and perimeter of a circle.

2. Use of Operators

- a. Write a Java program to add two binary numbers.
- b. Write a Java program to convert a decimal number to binary number and vice versa.
- c. Write a Java program to reverse a string.

3. Java Data Types

- a. Write a Java program to count the letters, spaces, numbers and other characters of an input string.
- 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.
- c. Find the smallest and largest element from the array

4. Methods and Constructors

- a. Designed a class SortData that contains the method asc() and desc().
- b. Designed a class that demonstrates the use of constructor and destructor.
- c. Write a java program to demonstrate the implementation of abstract class.

5. Inheritance

- a. Write a java program to implement single level inheritance.
- b. Write a java program to implement method overriding
- c. Write a java program to implement multiple inheritance.

6. Packages and Arrays

- a. Create a package, Add the necessary classes and import the package in java class.
- b. Write a java program to add two matrices and print the resultant matrix.
- c. Write a java program for multiplying two matrices and print the product for the same.

7. Exception Handling and Multithreading

- a. Write a java program to implement exception handling.
- b. Write a java program to implement thread life cycle.
- c. Write a java program to implement multithreading.

8. File Handling

- a. Write a java program to open a file and display the contents in the console window.
- b. Write a java program to copy the contents from one file to other file.
- c. Write a java program to read the student data from user and store it in the file.

9. Mini Project using Java AWT / Applet and Any Database.

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Course Name: Applied Mathematics | | Course Code: UGIT302 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: Understanding integration techniques as well as applications of integrals, parametric curves in a plane, as well as infinite sequences and series.

Expected Learning Outcomes:

- Compute a given integral using the most efficient method;
- Use integrals to formulate and solve application problems in science and engineering;
- Construct and plot parametric and polar curves;
- Identify different types of series and determine whether a particular series converges;
- Find the interval of convergence of a power series;
- Apply Taylor series to approximate functions and estimate the error of approximation.

| Unit | Details | Lectures |
|-------------|---|-----------------|
| 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 | 10 |

| | | |
|------------|---|-----------|
| | of matrix to a diagonal matrix which has elements as characteristics values. | |
| II | <p>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)</p> | 10 |
| III | <p>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</p> <p>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.</p> <p>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 expression 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</p> | 10 |
| IV | <p>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</p> <p>The 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, Laplace Transformation of Special Function, Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function (Unit Impulse Function)</p> | 10 |

| | | |
|-----------|---|-----------|
| V | <p>Multiple Integrals: Double Integral, Change of the order of the integration, Double integral in polar co-ordinates, Triple integrals.</p> <p>Applications of integration: Areas, Volumes of solids.</p> | 10 |
| VI | <p>Beta and Gamma Functions: Definitions, Properties and Problems. Duplication formula.</p> <p>Differentiation Under the Integral Sign : Introduction , Leibnitz Theorem for DUIS</p> <p>Error Functions: Introduction, Definitions of Error Functions, Properties of Error Functions.</p> | 10 |

| Books and References: | | | |
|------------------------------|--|-----------------------------------|---------------------|
| Sr. No. | Title | Author/s | Publisher |
| 1. | A text book of Applied Mathematics Vol I | P. N. Wartikar and J. N. Wartikar | Pune VidyathiGraha |
| 2. | Applied Mathematics II | P. N. Wartikar and J. N. Wartikar | Pune VidyathiGraha |
| 3. | Higher Engineering Mathematics | Dr. B. S. Grewal | Khanna Publications |

| B. Sc (Information Technology) | | Semester – III | |
|---|------------------------------|-----------------------------|--------------|
| Course Name: Applied Mathematics Practical | | Course Code: UGIT3P2 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

| List of Problem Solving: | |
|---------------------------------|--|
| 1 | Problem Solving based on Matrices |
| 2 | Problem Solving based on Complex Numbers |
| 3 | Problem Solving based on Equation of the first order and of the first degree |

| | |
|----|---|
| 4 | Problem Solving based on Differential equation of the first order of a degree higher than the first |
| 5 | Problem Solving based on Linear Differential Equations with Constant Coefficients |
| 6 | Problem Solving based on The Laplace Transform |
| 7 | Problem Solving based on Inverse Laplace Transform |
| 8 | Problem Solving based on Multiple Integrals |
| 9 | Problem Solving based on Applications of integration |
| 10 | Problem Solving based on Beta and Gamma Functions |
| 11 | Problem Solving based on Differentiation Under the Integral Sign |
| 12 | Problem Solving based on Error Functions |

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Course Name: Linux System Administration | | Course Code: UGIT303 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives:

- To demonstrate proficiency with the Linux command line interface, directory & file management techniques, file system organization, and tools commonly found on most Linux distributions.
- To effectively operate a Linux system inside of a network environment to integrate with existing service solutions.
- To demonstrate the ability to troubleshoot challenging technical problems typically encountered when operating and administering Linux systems.

Expected Learning Outcomes:

- Student will be able to understand and implement command line and system administration tasks.
- Student will be able to understand configuration and management of storage, network and managing user accounts.
- Student will be able to understand and implement security in the server, setting up cryptographic services and file server.
- Student will be able to configure DNS, DHCP and set up the Mail Server.
- Student will be able to implement the bash shell scripting and configuring Red Hat Enterprise Linux.
- Student will be able to understand and implement clustering and configure a network server as the installation server.

| Unit | Details | Lectures |
|-------------|--|-----------------|
| I | <p>Introduction to Red Hat Enterprise Linux: Linux, Open Source and Red Hat, Origins of Linux, Distributions, Duties of Linux System Administrator.</p> <p>Command Line : Working with the Bash Shell, Getting the Best of Bash, Useful Bash Key Sequences, Working with Bash History, Performing Basic File System Management Tasks, Working with Directories, Piping and Redirection, Finding Files</p> <p>System Administration Tasks : Performing Job Management Tasks, System and Process Monitoring and Management, Managing Processes with ps, Sending Signals to Processes with the kill Command, using top to Show Current System Activity, Managing</p> | 10 |

| | | |
|-----|---|----|
| | <p>Process Niceness, Scheduling Jobs, Mounting Devices, Working with Links, Creating Backups, Managing Printers, Setting Up System Logging, Setting Up Rsyslog, Common Log Files, Setting Up Logrotate</p> <p>Managing Software: Understanding RPM, Understanding Meta Package Handlers, Creating Your Own Repositories, Managing Repositories, Installing Software with Yum, Querying Software, Extracting Files from RPM Packages</p> | |
| II | <p>Configuring and Managing Storage : Understanding Partitions and Logical Volumes, Creating Partitions, Creating File Systems, File Systems Overview, Creating File Systems, Changing File System Properties, Checking the File System Integrity, Mounting File Systems Automatically Through fstab, Working with Logical Volumes, Creating Logical Volumes, Resizing Logical Volumes, Working with Snapshots, Replacing Failing Storage Devices, Creating Swap Space, Working with Encrypted Volumes</p> <p>Connecting to the Network : Understanding Network Manager, Working with Services and Run levels, Configuring the Network with Network Manager, Working with system-config-network, Network Manager Configuration Files, Network Service Scripts, Networking from the Command Line, Troubleshooting Networking, Setting Up IPv6, Configuring SSH, Enabling the SSH Server, Using the SSH Client, Using PuTTY on Windows Machines, Configuring Key Based SSH Authentication, Using Graphical Applications with SSH, Using SSH Port Forwarding, Configuring VNC Server Access .</p> <p>Working with Users, Groups, and Permissions : Managing Users and Groups, Commands for User Management, Managing Passwords, Modifying and Deleting User Accounts, Configuration Files, Creating Groups, Using Graphical Tools for User, and Group Management, Using External Authentication Sources, the Authentication Process, sssd, nsswitch, Pluggable Authentication Modules, Managing Permissions, the Role of Ownership, Basic Permissions: Read, Write, and Execute, Advanced Permissions, Working with Access Control Lists, Setting Default Permissions with umask, Working with Attributes</p> | 10 |
| III | <p>Securing Server with iptables : Understanding Firewalls, Setting Up a Firewall with system-config-firewall, Allowing Services, Trusted Interfaces, Masquerading, Configuration Files, Setting Up a Firewall with iptables, Tables, Chains, and Rules, Composition of Rule, Configuration Example, Advanced iptables Configuration, Configuring Logging, The Limit Module, Configuring NAT</p> <p>Setting Up Cryptographic Services: Introducing SSL, Proof of Authenticity: The Certificate Authority, Managing Certificates with</p> | 10 |

| | | |
|----|--|----|
| | <p>openssl, Creating a Signing Request, Working with GNU Privacy Guard, Creating GPG Keys, Key Transfer, Managing GPG Keys, Encrypting Files with GPG, GPG Signing, Signing RPM Files Configuring</p> <p>Server for File Sharing :What is NFS? Advantages and Disadvantages of NFS, Configuring NFS4, Setting Up NFSv4, Mounting an NFS Share, Making NFS Mounts Persistent, Configuring Automount, Configuring Samba, Setting Up a Samba File Server, Samba Advanced Authentication Options, Accessing Samba Shares, Offering FTP Services</p> | |
| IV | <p>Configuring DNS and DHCP : Introduction to DNS, The DNS Hierarchy, DNS Server Types, The DNS Lookup Process, DNS Zone Types, Setting Up a DNS Server, Setting Up a Cache-Only Name Server, Setting Up a Primary Name Server, Setting Up a Secondary Name Server, Understanding DHCP, Setting Up a DHCP Server</p> <p>Setting Up a Mail Server : Using the Message Transfer Agent, the Mail Delivery Agent, the Mail User Agent, Setting Up Postfix as an SMTP Server, Working with Mutt, Basic Configuration, Internet Configuration, Configuring Dovecot for POP and IMAP</p> | 10 |
| V | <p>Configuring Apache on Red Hat Enterprise Linux : Configuring the Apache Web Server, creating a Basic Website, Understanding the Apache Configuration Files, Apache Log Files, Working with Virtual Hosts, Securing the Web Server with TLS Certificates, Configuring Authentication, Setting Up Authentication with .htpasswd, Configuring LDAP Authentication, Setting Up MySQL</p> <p>Introducing Bash Shell Scripting: Introduction, Elements of a Good Shell Script, Executing the Script, Working with Variables and Input, Understanding Variables, Variables, Subshells, and Sourcing, Working with Script Arguments, Asking for Input, Using Command Substitution, Substitution Operators, Changing Variable Content with Pattern Matching, Performing Calculations, Using Control Structures, Using if...then...else, Using case, Using while, Using until, Using for, Configuring bootingwith GRUB.</p> | 10 |
| VI | <p>High-Availability Clustering : High-Availability Clustering, The Workings of High Availability, High-Availability Requirements, Red Hat High-Availability Add-on Software, Components, Configuring Cluster-Based Services, Setting Up Bonding, Setting Up Shared Storage, Installing the Red Hat High Availability Add-On, Building the Initial State of the Cluster, Configuring Additional Cluster Properties, Configuring a Quorum Disk, Setting Up Fencing, Creating Resources and Services, Troubleshooting a Nonoperational Cluster, Configuring GFS2 File</p> <p>Systems Setting Up an Installation Server: Configuring a Network Server as an Installation Server, Setting Up a TFTP and DHCP Server</p> | 10 |

| | | |
|--|---|--|
| | for PXE Boot, Installing the TFTP Server, Configuring DHCP for PXE Boot, Creating the TFTP PXE Server Content, creating a Kickstart File, Using a Kickstart File to Perform an Automated, Installation, Modifying the Kickstart File with, system-config-kickstart, Making Manual Modifications to the Kickstart File | |
|--|---|--|

| Books and References: | | | | |
|------------------------------|--|------------------------------|---------------------|-----------------|
| Sr. No. | Title | Author/s | Publisher | Edition |
| 1. | Red Hat Enterprise Linux 6 Administration | Sander van Vugt | John Wiley and Sons | |
| 2. | Red hat Linux Networking and System Administration | Terry Collings and Kurt Wall | Wiley | 3 rd |
| 3. | Linux Administration: A Beginner's Guide | Wale Soyinka | TMH | 5 th |

| B. Sc (Information Technology) | | Semester – III | |
|---|------------------------------|-----------------------------|--------------|
| Course Name: Linux System Administration Practical | | Course Code: UGIT3P3 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

| List of Practical: | |
|---------------------------|--|
| 0. | Installation of RHEL 6.X |
| 1. | Graphical User Interface and Command Line Interface and Processes |
| a. | Exploring the Graphical Desktop |
| b. | The Command Line Interface |
| c. | Managing Processes |
| | |
| 2. | Storage Devices and Links, Backup and Repository |
| a. | Working with Storage Devices and Links |
| b. | Making a Backup |
| c. | Creating a Repository |
| | |
| 3. | Working with RPMsm Storage and Networking |
| a. | Using Query Options |
| b. | Extracting Files From RPMs |
| c. | Configuring and Managing Storage |
| d. | Connecting to the Network |
| | |
| 4. | Working with Users, Groups, and Permissions |
| | |

| | |
|------------|---|
| 5. | Firewall and Cryptographic services |
| a. | Securing Server with iptables |
| b. | Setting Up Cryptographic Services |
| | |
| 6. | Configuring Server for File Sharing |
| a. | Configuring NFS Server and Client |
| b. | Configuring Samba |
| c. | Configuring FTP |
| | |
| 7. | DNS, DHCP and Mail Server |
| a. | Configuring DNS |
| b. | Configuring DHCP |
| c. | Setting Up a Mail Server |
| | |
| 8. | Web Server |
| a. | Configuring Apache on Red Hat Enterprise Linux |
| b. | Writing a Script to Monitor Activity on the Apache Web Server |
| | Using the select Command |
| | |
| 9. | Shell Scripts and High-Availability Clustering |
| a. | Writing Shell Scripts |
| b. | Configuring Booting with GRUB |
| c. | Configuring High Availability Clustering |
| | |
| 10. | Setting Up an Installation Server |
| a. | Configuring Network Server as an Installation Server |
| b. | Setting Up a TFTP and DHCP Server for PXE Boot |

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Course Name: Database Management Systems | | Course Code: UGIT304 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives:

To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage.

Expected Learning Outcomes:

- Student will be able to understand and implement database transactions and data models.
- Student will be able to understand and implement ER Diagram and Unified Modeling Language.
- Student will be able to understand the integrity rules.
- Student will be able to understand and implement Normalization
- Student will be able to understand and implement Relational Algebra and Calculus Relational Algebra.
- Student will be able to implement Constraints, Views and SQL.
- Student will be able to understand ACID Properties and implement concurrency control using Deadlocks.
- Student will be able to understand and implement PL-SQL.

| Unit | Details | Lectures |
|-------------|--|-----------------|
| I | Introduction to Databases and Transactions : What is database system, purpose of database system, view of data, Relationaldatabases, database architecture, transaction management Data Models : The importance of data models, Basic building blocks, Business rules, The evolutionof data models, Degrees of data abstraction. | 10 |
| II | Database Design,ER Diagram and Unified Modeling Language : Database design and ER Model:overview, ERModel, | 10 |

| | | |
|------------|--|-----------|
| | Constraints, ER Diagrams, ER Issues, weak entity sets, Codd's rules, Relational Schemas, Introduction to UML Relational database model: Logical view of data, keys, integrity rules, Relational Database design: features of good relational database design, atomic domain and Normalization (1NF, 2NF, 3NF, BCNF). | |
| III | Relational Algebra and Calculus Relational algebra: Introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, and semantics. Operators, grouping and ungrouping, relational comparison Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs. algebra, computational capabilities | 10 |
| IV | Constraints, Views and SQL : Constraints, types of constraints, Integrity constraints, Views: Introduction to views, data independence, security, updates on views, comparison between tables and views SQL: data definition, aggregate function, Null Values, nested sub queries, Joined relations. Triggers. | 10 |
| V | Transaction management and Concurrency: Control Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management. | 10 |
| VI | PL-SQL: Beginning with PL / SQL, Identifiers and Keywords, Operators, Expressions, Sequences, Control Structures, Cursors and Transaction, Collections and composite data types, Procedures Functions, Exceptions Handling, Packages, With Clause and Hierarchical Retrieval, Triggers. | 10 |

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|---------------------------------------|--------------------------------------|------------------|-----------------|------|
| 1. | Database System and Concepts | A Silberschatz, H Korth, S Sudarshan | McGraw-Hill | Fifth Edition | |
| 2. | Database Systems | Rob Coronel | Cengage Learning | Twelfth Edition | |
| 3. | Programming with PL/SQL for Beginners | H.Dand, R.Patil and T. Sambare | X –Team | First | 2011 |
| 4. | Introduction to Database System | C.J.Date | Pearson | First | 2003 |

| | | | |
|---|--|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Course Name: Database Management Systems Practical | | Course Code: UGIT3P4 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |

| | | | |
|-------------------|-----------------------|----|----|
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

| | |
|-----------|--|
| 1. | SQL Statements – 1 |
| a. | Writing Basic SQL SELECT Statements |
| b. | Restricting and Sorting Data |
| c. | Single-Row Functions |
| 2. | SQL Statements – 2 |
| a. | Displaying Data from Multiple Tables |
| b. | Aggregating Data Using Group Functions |
| c. | Sub queries |
| 3. | Manipulating Data |
| a. | Using INSERT statement |
| b. | Using DELETE statement |
| c. | Using UPDATE statement |
| 4. | Creating and Managing Tables |
| a. | Creating and Managing Tables |
| b. | Including Constraints |
| 5. | Creating and Managing other database objects |
| a. | Creating Views |
| b. | Other Database Objects |
| c. | Controlling User Access |
| 6. | Using SET operators, Date/Time Functions, GROUP BY clause (advanced features) and advanced subqueries |
| a. | Using SET Operators |
| b. | Datetime Functions |
| c. | Enhancements to the GROUP BY Clause |
| d. | Advanced Subqueries |
| 7. | PL/SQL Basics |
| a. | Declaring Variables |
| b. | Writing Executable Statements |
| c. | Interacting with the Oracle Server |
| d. | Writing Control Structures |
| 8. | Composite data types, cursors and exceptions. |
| a. | Working with Composite Data Types |
| b. | Writing Explicit Cursors |
| c. | Handling Exceptions |

| | |
|------------|-----------------------------------|
| 9. | Procedures and Functions |
| a. | Creating Procedures |
| b. | Creating Functions |
| c. | Managing Subprograms |
| d. | Creating Packages |
| 10. | Creating Database Triggers |

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|---------------------------------------|--------------------------------------|-------------|---------------|------|
| 1. | Database System and Concepts | A Silberschatz, H Korth, S Sudarshan | McGraw-Hill | Fifth Edition | |
| 2. | Programming with PL/SQL for Beginners | H.Dand , R.Patil and T. Sambare | X –Team | First | 2011 |
| 3. | PL/SQL Programming | Ivan Bayross | BPB | First | 2010 |

| B. Sc (Information Technology) | | Semester – III | |
|--|---------------------------|-----------------------------|--------------|
| Course Name: Data Structures | | Course Code: UGIT305 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives:

- To understand the abstract data types stack, queue, deque, and list.
- To understand the performance of the implementations of basic linear data structures.

Expected Learning Outcomes:

- Understanding prefix, infix, and postfix expression formats.
- Student will be able to understand and implement Arrays.
- Student will be able to understand and implement Linked List.
- Student will be able to understand and implement Doubly-Linked List.
- Student will be able to understand and implement Stack and Queues.
- Student will be able to understand and implement Sorting and Searching techniques.
- Student will be able to understand and implement Trees and advance tree structures.
- Student will be able to understand Hashing.
- Student will be able to understand and implement Graphs.

| 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 | 10 |

| | | |
|-----|--|----|
| | <p>Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation.</p> <p>Array: Introduction, One Dimensional Array, Memory Representation of One Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multi-Dimensional Arrays, Sparse Arrays, SparseMatrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays.</p> | |
| II | <p>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,</p> | 10 |
| III | <p>Doubly 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.</p> | 10 |
| IV | <p>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.</p> <p>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.</p> | 10 |
| V | <p>Sorting and Searching Techniques: Bubble, Selection, Insertion, Merge Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search.</p> <p>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.</p> <p>Advanced Tree Structures: Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, and Operations performed on AVL Tree, 2-3 Tree, and B-Tree.</p> | 10 |

| | | |
|-----------|--|-----------|
| VI | <p>Hashing Techniques: Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Buckethashing, Deletion and rehashing</p> <p>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.</p> | 10 |
|-----------|--|-----------|

| Books and References: | | | | | |
|------------------------------|---|---|---------------------|----------------|-------------|
| Sr. No. | Title | Author/s | Publisher | Edition | Year |
| 1 | A Simplified Approach to Data Structures | Lalit Goyal, Vishal Goyal, Pawan Kumar | SPD | 1 st | 2014 |
| 2 | An Introduction to Data Structure with Applications | Jean – Paul Tremblay and Paul Sorenson | Tata MacGraw Hill | 2 nd | 2007 |
| 3 | Data Structure and Algorithm | Maria Rukadikar | SPD | 1 st | 2017 |
| 4 | Schaum's Outlines Data structure | Seymour Lipschutz | Tata McGraw Hill | 2 nd | 2005 |
| 5 | Data structure – A Pseudocode Approach with C | AM Tanenbaum, Y Langsam and MJ Augustin | Prentice Hall India | 2 nd | 2006 |
| 6 | Data structure and Algorithm Analysis in C | Weiss, Mark Allen | Addison Wesley | 1 st | 2006 |

| B. Sc (Information Technology) | | Semester – III | |
|--|------------------------------|-----------------------------|--------------|
| Course Name: Data Structures Practical | | Course Code: UGIT3P5 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

| List of Practical : | |
|----------------------------|---|
| 1. | Implement the following: |
| 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] |

| | |
|-----------|---|
| b. | Read the two arrays from the user and merge them and display the elements in sorted order.[Menu Driven] |
| c. | Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven] |
| | |
| 2. | Implement the following for Linked List: |
| 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 |
| c. | Write a program to create double linked list and sort the elements in the linked list |
| | |
| 3. | Implement the following for Stack: |
| a. | Write a program to implement the concept of Stack with Push, Pop, Display and Exit operations |
| b. | Write a program to convert an infix expression to postfix and prefix conversion. |
| c. | Write a program to implement Tower of Hanoi problem |
| | |
| 4 | Implement the following for Queue: |
| a. | Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations. |
| b. | Write a program to implement the concept of Circular Queue |
| c. | Write a program to implement the concept of Deque. |
| | |
| 5 | Implement the following sorting techniques: |
| a. | Write a program to implement bubble sort. |
| b. | Write a program to implement selection sort. |
| c. | Write a program to implement insertion sort. |
| 6 | Implement the following data structure techniques: |
| a. | Write a program to implement merge sort. |
| b. | Write a program to search the element using sequential search. |
| c. | Write a program to search the element using binary search. |
| | |
| 7 | Implement the following data structure techniques: |
| a. | Write a program to create the tree and display the elements. |
| b. | Write a program to construct the binary tree. |
| c. | Write a program for inorder, postorder and preorder traversal of tree |
| | |
| 8 | Implement the following data structure techniques: |
| a. | Write a program to insert the element into maximum heap. |
| b. | Write a program to insert the element into minimum heap. |
| | |
| 9 | Implement the following data structure techniques: |
| a. | Write a program to implement the collision technique. |
| b. | Write a program to implement the concept of linear probing. |
| | |
| 10 | Implement the following data structure techniques: |

| | |
|----|---|
| a. | Write a program to generate the adjacency matrix. |
| b. | Write a program for shortest path diagram. |

| | | | |
|--|---------------------------|-----------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Elective Course Name: System Software | | Course Code: UGIT306 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives:

- To understand operating systems and application systems.
- Using the utility programs.

Expected Learning Outcomes:

- Student will be able to distinguish between operating systems software and application systems software.
- Student will be able to describe commonly used operating systems.
- Student will be able to identify the primary functions of an Operating System.

| <ul style="list-style-type: none"> • Student will be able to describe the “boot” process. • Student will be able to identify Desktop and Windows features. • Student will be able to use Utility programs. • Student will be able to discuss the pros and cons of the three major operating systems. | | |
|--|---|----------|
| Unit | Details | Lectures |
| I | Language Processor: Fundamentals of Language Processing and language Specification, Grammar and Types of Grammar, Toy Compiler, Introduction to Data structures: Heap and heap allocation, sorting methods. | 10 |
| II | Scanning and Parsing: Scanning, Finite Automata: DFA and NFA, Conversion of NFA into DFA, Top Down Parsing, Bottom up Parsing, Introduction to LEX and YACC tools. | 10 |
| III | Assemblers: Elements of Assembly Language Programming, a Simple Assembly Language Scheme, Pass Structure of Assembler, Design of Two Pass Assembler | 10 |
| IV | Macro Processors: Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Preprocessor, Design of Macro Preprocessor: Single Pass Algorithm, Two Pass Algorithm and Macro Calls within Macro Calls. | 10 |
| V | Loader and Linkers: Relocation And Linking Concept, Design of a Linker, Various schemes of Loader. Introduction to Compilers: Aspects of Compilation, Memory Allocation, Compilation of Expression, Compilation of Control Structure, Code Optimization, Interpreters. | 10 |
| VI | Editors and Debuggers: Various types of Editors and Debuggers and its design. Device Drivers: Introduction to the Device Driver, Requirements of Device Driver, Types of Device Driver. | 10 |

| Books and References: | | | | | |
|-----------------------|---|----------------------------------|-------------------------|---------|------|
| Sr. No. | Title | Author/s | Publisher | Edition | Year |
| 1 | Introduction to System Software | Dhamdhare | McGraw Hill | | |
| 2 | Compilers, Principles, Techniques and Tools | Aho. A.V. Sethi R. and Ulman J.D | Pearson | | |
| 3 | Systems Programming | Srimanta Pal | Oxford University Press | | |
| 4 | System Programming | John Donovan | TMH | | |
| 5 | Software-An Introduction to | Leland L. Beck and D. Manjula | Pearson | | |

| | | | | | |
|---|-------------------------------------|-----|-----|--|--|
| | Systems Programming | | | | |
| 6 | Compiler Design Using Flex and Yacc | Das | PHI | | |

| | | | |
|--|------------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – III | |
| Elective Course Name: System Software Practical | | Course Code: UGIT3P6 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

Practical list -1

1. Packet Monitoring software (tcpdump, snort, ethereal)
2. Trace route, Ping, Finger, Nmap
3. Server configuration (FTP, SMTP, DNS)
4. NFS Configuration
5. Firewall Configuration using iptables/ipchains (Linux only)
6. Experiments using Turbo C Assembler

All the above experiments may be performed in both Unix /Linux & Windows

Practical list -2

1. Assignments on class, constructor, overloading, inheritance, overriding
2. Assignments on wrapper class, vectors, arrays
3. Assignments on developing interfaces- multiple inheritance, extending interfaces
4. Assignments on creating and accessing packages
5. Assignments on multithreaded programming, handling errors and exceptions, applet programming and graphics programming
6. Use of CASE tools

Note: Use Java as programming language.

Semester – IV

| | | | |
|---|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – IV | |
| Course Name: Computer Graphics and Animation | | Course Code: UGIT401 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them. Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.

Expected Learning Outcomes:

- Student will be able to list the basic concepts used in computer graphics.
- Student will be able to implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
- Student will be able to understand and implement 2 dimensional transformations.
- Student will be able to understand and implement 3 dimensional transformations.
- Student will be able to understand and implement curve.
- Student will be able to understand Quadric Surfaces, Bezier Surfaces, Fractals, Fractal lines and Surfaces.
- Student will be able to describe the importance of viewing and projections.

| Unit | Details | Lectures |
|------|--|----------|
| I | <p>Introduction to Computer Graphics: 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 display</p> <p>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, Cohen-Sutherland and Liang-Barsky, Clipping Polygons problem with multiple components. ,polygon filling methods: Seed fill, Scan Line, Flood fill and Boundary fill.</p> | 10 |
| II | <p>Two-Dimensional Transformations: 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.</p> | 10 |
| III | <p>Three-Dimensional Transformations: 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,</p> | 10 |

| | | |
|-----------|---|-----------|
| | Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections. | |
| IV | <p>Viewing in 3D : 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.</p> <p>Light: Radiometry, Transport, Equation, Photometry.</p> <p>Color: Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance</p> | 10 |
| V | <p>Visible-Surface Determination: 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.</p> <p>Plane Curves and Surfaces: 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. Fractals, Fractal lines and Surfaces.</p> | 10 |
| VI | <p>Computer Animation : Principles of Animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects.</p> <p>Image Manipulation and Storage: 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.</p> | 10 |

| Books and References: | | | | | |
|------------------------------|---|--|------------------|----------------|-------------|
| Sr. No. | Title | Author/s | Publisher | Edition | Year |
| 1. | Computer Graphics - Principles and Practice | J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes | Pearson | 2nd | |
| 2. | Steve Marschner, Peter Shirley | Fundamentals of Computer Graphics | CRC press | 4th | 2016 |
| 3. | Computer Graphics | Hearn, Baker | Pearson | 2nd | |

| | | | | | |
|----|---|---|-----|-----|--|
| | | | | | |
| 4. | Principles of Interactive Computer Graphics | William M. Newman and Robert F. Sproull | TMH | 2nd | |
| 5. | Mathematical Elements for CG | D. F. Rogers, J. A. Adams | TMH | 2nd | |

| | | | |
|---|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – IV | |
| Course Name: Computer Graphics and Animation Practical | | Course Code: UGIT4P1 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2½ | 50 |
| | Internal | -- | -- |

| List of Practical | |
|--------------------------|--|
| 1. | Solve the following: |
| a. | Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them. |
| b. | Draw a co-ordinate axis at the center of the screen. |
| 2. | Solve the following: |
| a. | Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message. |
| b. | Draw a simple hut on the screen. |
| 3. | Draw the following basic shapes in the center of the screen : |
| a. | i. Circle ii. Rectangle iii. Square iv. Concentric Circles v. Ellipse vi. Line |
| 4. | Solve the following: |
| a. | Develop the program for DDA Line drawing algorithm. |
| b. | Develop the program for Bresenham's Line drawing algorithm. |
| 5. | Solve the following: |
| a. | Develop the program for the mid-point circle drawing algorithm. |
| b. | Develop the program for the mid-point ellipse drawing algorithm. |
| 6. | Solve the following: |
| a. | Write a program to implement 2D scaling. |
| b. | Write a program to perform 2D translation |

| | |
|------------|---|
| 7. | Solve the following: |
| a. | Perform 2D Rotation on a given object. |
| | |
| 8. | Solve the following: |
| a. | Write a program to implement Cohen-Sutherland clipping. |
| b. | Write a program to implement Liang - Barsky Line Clipping Algorithm |
| | |
| 9. | Solve the following: |
| a. | Write a program to fill a circle using Flood Fill Algorithm. |
| b. | Write a program to fill a circle using Boundary Fill Algorithm. |
| | |
| 10. | Solve the following: |
| a. | Develop a simple text screen saver using graphics functions. |
| b. | Perform smiling face animation using graphic functions. |
| c. | Draw the moving car on the screen. |

| | | |
|--|----------------------|--------------|
| B. Sc (Information Technology) | Semester – IV | |
| Course Name: Introduction to Embedded Systems | Course Code: UGIT402 | |
| Periods per week (1 Period is 50 minutes) | 5 | |
| Credits | 2 | |
| | Hours | Marks |

| | | | |
|-------------------|--------------------|----|----|
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: – The objective of this course is to make the learner ready to understand organization and microcontroller. To cover-up with the pre-requisites of embedded computing.

Expected Learning Outcomes:

- Student will be able to identify the characteristics and quality attributes of embedded systems
- Student will be able to understand and discuss different processors, microcontrollers, and programmable logic devices.
- Student will be able to identify sensors and actuators, embedded firmware and other system components.
- Student will be able to understand the concept of embedded hardware and peripherals.
- Student will be able to write programs in C for 8051 Microcontroller.
- Student will be able to design and implement embedded system using 8051 microcontroller.

| Unit | Details | Lectures |
|------|---|----------|
| I | <p>Introduction: Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems.</p> <p>Characteristics and quality attributes of embedded systems: Characteristics of Embedded Systems, operational and non-operational quality attributes of Embedded Systems</p> | 10 |
| II | <p>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.</p> <p>Embedded Systems – Application and Domain Specific: Application specific – washing machine, domain specific - automotive.</p> | 10 |
| III | <p>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.</p> <p>Peripherals: Control and Status Registers, Device Driver, Timer Driver - Watchdog Timers.</p> | 10 |
| IV | <p>The 8051 Microcontrollers : Microcontrollers and Embedded processors, Overview of 8051 family.8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory.</p> <p>8051 Programming in C: Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs.</p> | 10 |

| | | |
|-----------|---|-----------|
| V | <p>Designing Embedded System with 8051 Micro- controller: Factors to be considered in selecting a controller, why 8051 Microcontroller, Designing with 8051.</p> <p>Programming embedded systems: Structure of embedded program, infinite loop, compiling, linking and debugging.</p> | 10 |
| VI | <p>Real Time Operating System (RTOS): Operating system basics, types of operating systems, Real-Time Characteristics, Selection Process of an RTOS.</p> <p>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</p> | 10 |

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|---|---------------------|------------------|---------|------|
| 1. | Programming Embedded Systems in C and C++ | Michael Barr | O'Reilly | First | 1999 |
| 2. | Introduction to embedded systems | Shibu K V | Tata Mcgraw-Hill | First | 2012 |
| 3. | The 8051 Microcontroller and Embedded Systems | Muhammad Ali Mazidi | Pearson | Second | 2011 |
| 4. | Embedded Systems | Rajkamal | Tata Mcgraw-Hill | | |

| | | | |
|--|------------------------------|-----------------------------|--------------|
| Course Name: Introduction to Embedded Systems Practical | | Course Code: UGIT4P2 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

List of Practicals:

| | |
|-------|--|
| 1) | Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects. a. Programming b. Execution c. Debugging |
| 2.A) | Configure timer control registers of 8051 and develop a program to generate given time delay. |
| 2.B) | To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them. |
| 3.A) | 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 |
| 3.B) | Interfacing LCD display with AT89S52. |
| 4.A) | Interfacing program for Matrix keyboard. |
| 4.B) | To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay. |
| 5.A) | Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope. |
| 5.B) | Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051. |
| 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. |
| 7) | Generate traffic signal. |
| 8) | Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope. |
| 9) | Implement Elevator control. |
| 10) | Using FlashMagic: |

| | |
|----|--|
| A) | To demonstrate the procedure for flash programming for reprogrammable embedded system board using FlashMagic |
| 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. |

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – IV | |
| Course Name: Computer Oriented Statistical Techniques | | Course Code: UGIT403 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: – To demonstrate understanding of numerical and statistical methods in support of the analysis, design and application for problem solving in the field of information technology

Expected Learning Outcomes:

- Student will be able to recognize the error in the number generated by the solution.
- Student will be to compute solution of algebraic and transcendental equation by numerical methods.
- Student will able to implement Bisection method and Newton Rapshon method.
- Student will be able to apply method of interpolation and extrapolation for prediction.
- Student will be able to recognize elements and variable in statistics and summarize qualitative and quantitative data.
- Student will be able to calculate mean, median and mode for individual series.

| Unit | Details | Lectures |
|-------------|--|-----------------|
| I | 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. | 10 |
| II | 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 | 10 |

| | | |
|------------|---|-----------|
| | <p>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, Standardized coefficient of Variation able; Standard Scores, Software and Measures of Dispersion.</p> <p>Introduction to R: Basic syntax, data types, variables, operators, control statements, R-functions, R – Vectors, R – lists, R Arrays.</p> | |
| III | <p>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.</p> <p>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.</p> | 10 |
| IV | <p>Elementary Sampling Theory : Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Di and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory</p> <p>Statistical Estimation Theory: Estimation of Parameters, Unbiased Estimates, coefficient Estimates, coefficient Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error</p> | 10 |
| V | <p>Decision Theory: Statistical Decisions, Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, 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 coefficient Test serences, Involving Binomial Distributions.</p> <p>Statistics in R: mean, median, mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R. Perform the Linear Regression ,Least squares means, Linear Least Square Regression using R.</p> | 10 |
| VI | <p>Small Sampling Theory: Small Samples, Student's t Distribution,</p> | 10 |

| | | |
|--|---|--|
| | <p>Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma , Degrees of Freedom, The F Distribution.</p> <p>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, Coe Contingency, Correlation of Attributes, Additive Property of chi-square.</p> | |
|--|---|--|

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|--|---------------------------------------|-----------------------------|-------------------|------|
| 1. | STATISTICS | Murray R. Spiegel, Larry J. Stephens. | McGRAW – HILL INTERNATIONAL | FOURTH | |
| 2. | A Practical Approach using R | R.B. Patil, H.J. Dand and R. Bhavsar | SPD | 1 st | 2017 |
| 3. | FUNDAMENTAL OF MATHEMATICAL STATISTICS | S.C. GUPTA and V.K. KAPOOR | SULTAN CHAND and SONS | ELEVENTH REVISED | 2011 |
| 4. | MATHEMATICAL STATISTICS | J.N. KAPUR and H.C. SAXENA | S. CHAND | TWENTIETH REVISED | 2005 |

| B. Sc (Information Technology) | | Semester – IV | |
|--|------------------------------|-----------------------------|--------------|
| Course Name: Computer Oriented Statistical Techniques Practical | | Course Code: UGIT4P3 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

List of Practicals:

- Using R execute the basic commands, array, list and frames.
- Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operations.

| |
|--|
| 3. Using R Execute the statistical functions:mean, median, mode, quartiles, range,inter quartile range histogram |
| 4. Using R import the data from Excel / .CSV file and Perform the above functions. |
| 5. Using R import the data from Excel / .CSV file and Calculate the standarddeviation, variance, co-variance. |
| 6. Using R import the data from Excel / .CSV file and draw the skewness. |
| 7. Import the data from Excel / .CSV and perform the hypothetical testing. |
| 8. Import the data from Excel / .CSV and perform the Chi-squared Test. |
| 9. Using R perform the binomial and normal distribution on the data. |
| 10. Perform the Linear Regression using R. |
| 11. Compute the Least squares means using R. |
| 12. Compute the Linear Least Square Regression |

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – IV | |
| Course Name: Software Project Management | | Course Code: UGIT404 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: – To understand the fundamental principles of Software Project management & will also have a good knowledge of responsibilities of project manager and how to handle these.

Expected Learning Outcomes:

- Student will be able to define various software application domains and remember different process model used in software development.
- Student will be able to elaborate the Project Planning.
- Student will be able to explain needs for software specifications also they can classify different types of software requirements and their gathering techniques.
- Student will be able to convert the requirements model into the design model and demonstrate use of software and user interface design principles.
- Student will be able to distinguish among SCM and SQA and can classify different testing strategies and tactics and compare them.
- Student will be able to justify role of SDLC in Software Project Development.

| Unit | Details | Lectures |
|-------------|---|-----------------|
| I | Introduction to Software Project Management: Introduction, Why is Software Project Management Important? What is a Project? Software Projects versus Other Types of Project, Contract Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of | 10 |

| | | |
|---|---|--|
| | <p>Categorizing Software Projects, Project Charter, Stakeholders, Setting Objectives, The Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices.</p> <p>Project Evaluation and Programme Management : Introduction, Business Case, Project Portfolio Management, Evaluation of Individual Projects, Cost–benefit Evaluation Techniques, Risk Evaluation, Programme Management, Managing the Allocation of Resources within Programmes, Strategic Programme Management, Creating a Programme, Aids to Programme Management, Some Reservations about Programme Management, Benefits Management.</p> <p>An Overview of Project Planning :Introduction to Step Wise Project Planning, Step 0: Select Project, Step 1: Identify Project Scope and Objectives, Step 2: Identify Project Infrastructure, Step 3: Analyse Project Characteristics, Step 4: Identify Project Products and Activities, Step 5: Estimate Effort for Each Activity, Step 6: Identify Activity Risks, Step 7: Allocate Resources, Step 8: Review/Publicize Plan, Steps 9 and 10: Execute Plan/Lower Levels of Planning</p> | |
| <p style="text-align: center;">II</p> | <p>Selection of an Appropriate Project Approach: Introduction, Build or Buy? Choosing Methodologies and Technologies, Software Processes and Process Models, Choice of Process Models, Structure versus Speed of Delivery, The Waterfall Model, The Spiral Model, Software Prototyping, Other Ways of Categorizing Prototypes, Incremental Delivery, Atern/Dynamic Systems Development Method, Rapid Application Development, Agile Methods, Extreme Programming (XP), Scrum, Lean Software Development, Managing Iterative Processes, Selecting the Most Appropriate Process Model.</p> <p>Software Effort Estimation: Introduction, Where are the Estimates Done? Problems with Over- and Under-Estimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Bottom-up Estimating, The Top-down Approach and Parametric Models, Expert Judgement, Estimating by Analogy, Albrecht Function Point</p> | <p style="text-align: center;">10</p> |
| <p style="text-align: center;">III</p> | <p>Activity Planning : Introduction, Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-on-Arrow Networks.</p> <p>Risk Management : Introduction, Risk, Categories of Risk, Risk Management Approaches, A Framework for Dealing with Risk, Risk Identification, Risk Assessment, Risk Planning, Risk Management, Evaluating Risks to the Schedule, Boehm’s Top 10 Risks and Counter Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts.</p> | <p style="text-align: center;">10</p> |

| | | |
|-----------|---|-----------|
| | Resource Allocation: Introduction, Nature of Resources, Identifying Resource Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resource Schedule, Cost Schedules, Scheduling Sequence. | |
| IV | Monitoring and Control: Introduction, Creating the Framework, Collecting the Data, Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Software Configuration Management (SCM). Managing Contracts : Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. | 10 |
| V | Managing People in Software Environments : Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham–Hackman Job Characteristics Model, Stress, Stress Management, Health and Safety, Some Ethical and Professional Concerns. Working in Teams : Introduction, becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership. | 10 |
| VI | Software Quality : Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models, ISO 9126, Product and Process Metrics, Product versus Process Quality Management, Quality Management Systems, Process Capability Models, Techniques to Help Enhance Software Quality, Testing, Software Reliability, Quality Plans. Project Closeout: Introduction, Reasons for Project Closure, Project Closure Process, Performing a Financial Closure, Project Closeout Report | 10 |

Books and References:

| Title | Author/s | Edition | Publisher |
|---|--|---------|-----------|
| Software Project Management | Bob Hughes, Mike Cotterell, Rajib Mall | TMH | 6th |
| Project Management and Tools & Technologies – An overview | Shailesh Mehta | SPD | 1st |
| Software Project Management | Walker Royce | Pearson | |

| | | | |
|---|------------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester IV | |
| Course Name: Software Project Management Practical | | Course Code: UGIT4P4 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |

| List of Practical (To be executed using Star UML or any similar software) | |
|--|---|
| 1. | Study and implementation of class diagrams. |
| 2. | Study and implementation of Use Case Diagrams. |
| 3. | Study and implementation of Entity Relationship Diagrams. |
| 4. | Study and implementation of Sequence Diagrams. |
| 5. | Study and implementation of State Transition Diagrams. |
| 6. | Study and implementation of Data Flow Diagrams. |
| 7. | Study and implementation of Collaboration Diagrams. |
| 8. | Study and implementation of Activity Diagrams. |
| 9. | Study and implementation of Component Diagrams. |
| | Study and implementation of Deployment Diagram |

| | |
|-----|--|
| 10. | |
|-----|--|

| | | | |
|--|---------------------------|-----------------------------|--------------|
| B. Sc (Information Technology) | | Semester – IV | |
| Course Name: Python Programming | | Course Code: UGIT405 | |
| Periods per week (1 Period is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory Examination | 2 | 60 |
| | Internal | -- | 40 |

Objectives: – The objective of this course is to introduce various concepts of programming to the students using Python.

Expected Learning Outcomes:

- Students will be able to understand the concepts of programming before actually starting to write new programs.
- Students will be able to understand what happens in the background when the programs are executed
- Students will be able to develop logic for Problem Solving.
- Students will be able to understand basic constructs of programming such as data, operations, conditions, loops, functions etc.
- Students should be able to apply the problem solving skills using syntactically simple language.
- Student will be able to implement objects and classes, multithreading and modules.

| | | |
|-------------|----------------|-----------------|
| Unit | Details | Lectures |
|-------------|----------------|-----------------|

| | | |
|-----|--|----|
| I | <p>Introduction: The Python Programming Language, History, features, Installing Python, Running Python program, Interactive Mode and Script Mode, Debugging : Syntax Errors, Runtime Errors, Semantic Errors, Experimental Debugging, The Difference Between Brackets, Braces, and Parentheses,</p> <p>Variables and Expressions: Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Order of Operations.</p> <p>Conditional Statements : if, if-else, nested if –else, Looping: for, while, nested loops</p> <p>Control statements : Terminating loops, skipping specific conditions</p> | 10 |
| II | <p>Functions: Why Functions? Function Calls, Type Conversion Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Fruitful Functions and Void Functions, Return Values, Incremental Development, Composition, Boolean Functions, More Recursion, Checking Types</p> | 10 |
| III | <p>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.</p> <p>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</p> | 10 |
| IV | <p>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 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</p> <p>Text Files: The File Object Attributes, Directories</p> <p>Built-in Exceptions: Handling Exceptions, Exception with Arguments, User-defined Exceptions</p> | 10 |
| V | <p>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</p> <p>Multithreaded Programming : Thread Module, creating a thread, synchronizing threads, multithreaded priority queue</p> <p>Modules : Importing module, Creating and exploring modules, Math module, Time module</p> | 10 |
| VI | <p>Creating the GUI Form and Adding Widgets: Button, Canvas, Check button, Entry, Frame, Label, Listbox, Menu button, Menu, Message, Radio button, Scale, Scrollbar, text, Toplevel, Spinbox, Paned Window, LabelFrame, tkMessageBox.</p> | 10 |

| | | |
|--|---|--|
| | <p>Widgets & Layout Management: Handling Standard attributes and Properties of Widgets ,</p> <p>Look and Feel Customization: Designing GUI applications with proper Layout Management features. Enhancing Look and Feel of GUI using different appearances of widgets.</p> <p>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.</p> | |
|--|---|--|

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|--|--|-----------------------|-----------------|------|
| 1. | Think Python | Allen Downey | O'Reilly | 1 st | 2012 |
| 2. | An Introduction to Computer Science using Python 3 | Jason Montojo, Jennifer Campbell, Paul Gries | SPD | 1 st | 2014 |
| 3. | Python GUI Programming Cookbook | Burkhard A. Meier | Packt | | 2015 |
| 4. | Introduction to Problem Solving with Python | E. Balagurusamy | TMH | 1 st | 2016 |
| 5. | Murach's Python programming | Joel Murach, Michael Urban | SPD | 1 st | 2017 |
| 6. | Object-oriented Programming in Python | Michael H. Goldwasser, David Letscher | Pearson Prentice Hall | 1 st | 2008 |
| 7. | Exploring Python | Budd | TMH | 1 st | 2016 |

| | | | |
|---|------------------------------|-----------------------------|--------------|
| B. Sc. (Information Technology) | | Semester – IV | |
| Elective Course Name: Python Programming Practical | | Course Code: UGIT4P5 | |
| Periods per week (1 Period is 50 minutes) | | 3 | |
| Credit: | | 2 | |
| | | Hours | Marks |
| Evaluation System | Practical Examination | 2½ | 50 |
| | Internal | -- | -- |
| | | | |

| List of Practical | |
|--------------------------|--|
| 1. | Write the program for the following: |
| 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. |

| | |
|-----------|---|
| 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. |
| c. | Write a program to generate the Fibonacci series. |
| d. | Write a function that reverses the user defined value. |
| e. | Write a function to check the input value is Armstrong and also write the function for Palindrome. |
| f. | Write a recursive function to print the factorial for a given number. |
| 2. | Write the program for the following: |
| 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. |
| b. | Define a function that computes the <i>length</i> of a given list or string. |
| c. | Define a <i>procedure</i> histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following: **** ***** ***** |
| 3. | Write the program for the following: |
| a. | A <i>pangram</i> is a sentence that contains all the letters of the English alphabet at least once, for example: <i>The quick brown fox jumps over the lazy dog</i> . Your task here is to write a function to check a sentence to see if it is a pangram or not. |
| b. | Take a list, say for example this one: a=[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] 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 |
| c. | Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers). i. Write a method called add which returns the sum of the attributes x and y. ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER. iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c. |

| | |
|-----------|--|
| | iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y. |
| 8. | Write the program for the following: |
| a. | <p>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.</p> <p>Now open a new file and save it in the same directory. You should now be able to import your own module like this:</p> <pre>import geometry</pre> <p>Try and add <code>print dir(geometry)</code> to the file and run it.</p> <p>Now write a function <code>pointyShapeVolume(x, y, squareBase)</code> that calculates the volume of a square pyramid if <code>squareBase</code> is True and of a right circular cone if <code>squareBase</code> is False. <code>x</code> is the length of an edge on a square if <code>squareBase</code> is True and the radius of a circle when <code>squareBase</code> is False. <code>y</code> is the height of the object. First use <code>squareBase</code> to distinguish the cases. Use the <code>circleArea</code> and <code>squareArea</code> from the <code>geometry</code> module to calculate the base areas.</p> |
| b. | Write a program to implement exception handling. |
| 9 | Mini Project using Python and Mysql |

Books and References:

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|--|--|-----------|-----------------|------|
| 1. | Think Python | Allen Downey | O’Reilly | 1 st | 2012 |
| 2. | An Introduction to Computer Science using Python 3 | Jason Montojo, Jennifer Campbell, Paul Gries | SPD | 1 st | 2014 |

| | | | |
|---|-----------------|-----------------------------|--------------|
| B.Sc.(Information Technology) | | Semester-IV | |
| Elective Course Name: Data warehousing | | Course Code: UGIT406 | |
| Periods per week(1 periods is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory | 2 | 60 |
| | Internal | --- | 40 |

Objectives: – This course will introduce the concepts of data ware house and data mining, which gives a complete description about the principles, used, architectures, applications, design and implementation of data mining and data ware housing concepts.

Expected Learning Outcomes:

- Students will be able to understand the concepts of Data Ware housing.
- Students will be able Explain the methodologies used for analysis of data
- Students should be able to describe various techniques which enhance the data modeling.
- Students should be made familiar about the basic constructs of programming such as data,operations, conditions, loops, functions etc.
- Students will be able to discuss various approaches with other techniques in data ware housing.
- Students will be able to compare various approaches with other techniques in data ware housing.

| Unit | Details | Lectures |
|------|--|----------|
| I | <p>Introduction to Data Warehousing: Introduction, Necessity, Framework of the datawarehouse, options, developing datawarehouses, end points.</p> <p>Data Warehousing Design Consideration and Dimensional Modeling: Defining Dimensional Model, Granularity of Facts, Additivity of Facts, Functional dependency of the Data, Helper Tables, Implementation many-to-many relationships between fact and dimensional modelling</p> | 10 |
| II | <p>An Introduction to Oracle Warehouse Builder: Installation of the database and OWB, About hardware and operating systems, Installing Oracle database software, Configuring the listener, Creating the database, Installing the OWB standalone software, OWB components and architecture, Configuring the repository and workspaces.</p> <p>Defining and Importing Source Data Structures: An overview of Warehouse Builder Design Center, Importing/defining source metadata, Creating a project, Creating a module, Creating an Oracle Database module, Creating a SQL Server database module, Importing source metadata from a database, Defining source metadata manually with the Data Object Editor, Importing source metadata from files.</p> | 10 |
| III | <p>Designing the Target Structure: Data warehouse design, Dimensional design, Cube and dimensions, Implementation of a dimensional model in a database, Relational implementation (star schema),Multidimensional implementation (OLAP),Designing the ACME data warehouse, Identifying the dimensions, Designing the cube, Data warehouse design in OWB, Creating a target user and module, Create a target user, Create a target module, OWB design objects.</p> <p>Creating the Target Structure in OWB: Creating dimensions in OWB, The Time dimension, Creating a Time dimension with the Time Dimension Wizard, The Product dimension, Product Attributes (attribute type),Product Levels, Product Hierarchy (highest to lowest),Creating the Product dimension with the New Dimension Wizard, The Store dimension, Store Attributes (attribute type), data type and size, and</p> | 10 |

| | | |
|-----------|--|-----------|
| | (Identifier),Store Levels, Store Hierarchy (highest to lowest),Creating the Store dimension with the New Dimension Wizard, Creating a cube in OWB, Creating a cube with the wizard, Using the Data Object Editor | |
| IV | <p>Extract, Transform, and Load Basics: ETL, Manual ETL processes, Staging, To stage or not to stage, Configuration of a staging area, Mappings and operators in OWB, The canvas layout, OWB operators, Source and target operators, Data flow operators, Pre/post-processing operators.</p> <p>Designing and building an ETL mapping: Designing our staging area, Designing the staging area contents, Building the staging area table with the Data Object Editor, Designing our mapping, Review of the Mapping Editor, Creating a mapping.</p> | 10 |
| V | <p>ETL: Transformations and Other Operators: STORE mapping, Adding source and target operators, Adding Transformation Operators, Using a Key Lookup operator, Creating an external table, Creating and loading a lookuptable, Retrieving the key to use for a Lookup Operator, Adding a Key Lookup operator, PRODUCT mapping, SALES cube mapping, Dimension attributes in the cube, Measures and other attributes in the cube, Mapping values to cube attributes, Mapping measures' values to a cube, Mapping PRODUCT and STORE dimension values to the cube, Mapping DATE_DIM values to the cube, Features and benefits of OWB.</p> <p>Validating, Generating, Deploying, and Executing Objects: Validating, Validating in the Design Center, Validating from the editors, Validating in the Data Object Editor, Validating in the Mapping, Editor, Generating, Generating in the Design Center, Generating from the editors, Generating inthe Data Object Editor, Generating in the Mapping Editor, Deploying, The Control Center Service, Deploying in the Design Center and Data Object Editor, The Control Center Manager, The Control Center Manager window overview, Deploying in the Control Center ,Manager, Executing, Deploying and executing remaining objects, Deployment Order, Execution order.</p> | 10 |
| VI | <p>Extra Features: Additional editing features, Metadata change management, Recycle Bin, Cut, copy, and paste, Snapshots, Metadata Loader (MDL) exports and imports, Synchronizing objects, Changes to tables, Changes to dimensional objects and auto-binding, Warehouse Builder online resources.</p> <p>Datawarehousing and OLAP: Defining OLAP, The Value of Multidimensional data, OLAP terminologies, Multidimensional architectures, Multidimensional views of relational data, Physical Multidimensional databases, Data Explosion, Integrated relational OLAP, Data sparsity and data explosion.</p> | 10 |

Books:

Data Warehousing by SoumendraMohanty, Tata McGrawHill

(Unit I: Chapter 1, 2 Unit VI: 14).

Oracle Warehouse Builder 11g, Getting Started by Bob Griesemer, PACKT Publishing, SPD.

(Unit II: Chapter 1,3 Unit III: Chapter 3,4 Unit IV: Chapter 5,6 Unit V: Chapter 7,8 Unit VI: Chapter 9)

References:

DW2.0 The architecture for Next Generation of Datawarehousing by W.H. Inmon, Derek Strauss, GeniaNeushloss, ELSEVIER.

Data Warehousing Fundamentals by PaulrajPonnian, John Wiley.

Building the data warehouse by, W.H.Inmon, third Edition, John Wiley.

The Data Warehouse Lifecycle toolkit by Ralph Kimball, John Wile

| | | | |
|---|-----------------|-----------------------------|--------------|
| B.Sc.(Information Technology) | | Semester-II | |
| Elective Course Name: Data warehousing Practical | | Course Code: UGIT4P6 | |
| Periods per week(1 periods is 50 minutes) | | 5 | |
| Credits | | 2 | |
| | | Hours | Marks |
| Evaluation System | Theory | 2½ | 50 |
| | Internal | --- | -- |

List of Practical

1. Importing the source data structures in Oracle.
2. Design the target data structure using Oracle
3. Create the target structure in OWB (Oracle Web Builder)
4. Designed and build the ETL mapping
5. Perform the ETL process and transform it to data marts.
6. Create the cube and process it in OWB.
7. Generate the different types of reports in using Oracle.
8. Perform the deployment of Warehouse
9. Create the Pivot table and Pivot chart using some existing data or create the new data.
10. Import the cube in access and create Pivot table and chart.