**THIRUVALLUVAR UNIVERSITY**

**VELLORE**

**MASTER OF COMPUTER APPLICATION**

**CBCS PATTERN**

(With effect from 2020 - 2021)

**The course of Study and scheme of Examination**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Study Components** | | | | | | | | | | | **Ins. Hrs./**  **Week** | | | | | **Credit** | | **Title of the Paper** | **Maximum Marks** | | |
| **Course Title** | | | | | | | | | | | **CIA** | **Uni. Exam** | **Total** |
| **SEMESTER 1** | | | | | | | | | | | | | | | | | | |  |
|  | Core | | | Paper -1 | | | | | | | | 5 | | | | 3 | | | Programming in C | 25 | 75 | 100 |
|  | Core | | | Paper -2 | | | | | | | | 5 | | | | 3 | | | Web Design | 25 | 75 | 100 |
|  | Core | | | Paper -3 | | | | | | | | 5 | | | | 3 | | | Data Structures | 25 | 75 | 100 |
|  | Practical | | | Paper -1 | | | | | | | | 3 | | | | 2 | | | Practical 1: Programming in C | 25 | 75 | 100 |
|  | Practical | | | Paper -2 | | | | | | | | 3 | | | | 2 | | | Practical 2: Web Design | 25 | 75 | 100 |
|  | Practical | | | Paper -3 | | | | | | | | 3 | | | | 2 | | | Practical 3: Data Structures using C | 25 | 75 | 100 |
| **Internal Elective for same major students** | | | | | | | | | | | | | | | | | | | | | | |
|  | Core Elective | | | Paper -1 | | | | | | | | 3 | | | | 3 | | | **(to choose one out of 3)**   1. Digital Logic Fundamentals 2. Computer Organization and Architecture 3. Fundamentals of Microprocessors | 25 | 75 | 100 |
| **External Major for other major Students (Inter/multi-disciplinary papers)** | | | | | | | | | | | | | | | | | | | | | | |
|  | Open Elective | | | | | | Paper-1 | | | | 3 | | | | 3 | | | | **(to choose one out of 3)**   1. E-Commerce 2. Introduction to Computer Application 3. Principles of Internet | 25 | 75 | 100 |
|  |  | | | | | |  | | | | **30** | | | | **21** | | | |  | **200** | **600** | **800** |
| **SEMESTER II** | | | | | | | | | | | | | | | | | | |  | **CIA** | **Uni. Exam** | **Total** |
|  | Core | | | | | | Paper-4 | | | 5 | | | | | | | 3 | | Programming with Java | 25 | 75 | 100 |
|  | Core | | | | | | Paper-5 | | | 4 | | | | | | | 3 | | Relational Database Management System | 25 | 75 | 100 |
|  | Core | | | | | | Paper-6 | | | 4 | | | | | | | 3 | | Open Source Technologies | 25 | 75 | 100 |
|  | Practical | | | | | | Paper-4 | | | 3 | | | | | | | 2 | | Practical 4: Programming with Java | 25 | 75 | 100 |
|  | Practical | | | | | | Paper-5 | | | 3 | | | | | | | 2 | | Practical 5: Relational Database Management System | 25 | 75 | 100 |
|  | Practical | | | | | | Paper-6 | | | 3 | | | | | | | 2 | | Practical 6: Open Source Technologies | 25 | 75 | 100 |
| **Internal Elective for same major students** | | | | | | | | | | | | | | | | | | | | | | |
|  | Core Elective | | | | | Paper-2 | | | | 3 | | | | | | 3 | | | **(to choose one out of 3)**   1. Operation Research 2. Graph Theory 3. Discrete Mathematics | 25 | 75 | 100 |
| **External Major for other major Students (Inter/multi-disciplinary papers)** | | | | | | | | | | | | | | | | | | | | | | |
|  | Open Elective | | | | Paper-2 | | | | 3 | | | | | | | | 3 | | **(to choose one out of 3)**   1. Problem Solving Techniques 2. Open Source Software 3. Principal of Web Design | 25 | 75 | 100 |
|  | \*Field Study | | | |  | | | | - | | | | | | | | 2 | |  | 100 | - | 100 |
|  | Compulsory Paper | | | |  | | | | 2 | | | | | | | | 2 | | Human Rights | 25 | 75 | 100 |
|  |  | | | |  | | | | **30** | | | | | | | | **25** | |  | **325** | **675** | **1000** |
|  |  | | | |  | | | | **120** | | | | | | | | **90** | |  |  |  |  |
| **SEMESTER III** | | | | | | | | | | | | | | | | | | |  | **CIA** | **Uni. Exam** | **Total** |
|  | Core | | Paper-7 | | | | | | | | | 5 | | | | | 3 | | Advance Java Programming | 25 | 75 | 100 |
|  | Core | | Paper-8 | | | | | | | | | 5 | | | | | 3 | | Unix and Shell Programming | 25 | 75 | 100 |
|  | Core | | Paper-9 | | | | | | | | | 5 | | | | | 3 | | Desktop Applications using C# | 25 | 75 | 100 |
|  | Practical | | Paper-7 | | | | | | | | | 3 | | | | | 2 | | Practical 7: Advance Java Programming | 25 | 75 | 100 |
|  | Practical | | Paper-8 | | | | | | | | | 3 | | | | | 2 | | Practical 8: Unix and Shell Programming | 25 | 75 | 100 |
|  | Practical | | Paper-9 | | | | | | | | | 3 | | | | | 2 | | Practical 9: Desktop Applications using C# | 25 | 75 | 100 |
| **Internal Elective for same major students** | | | | | | | | | | | | | | | | | | | | | | |
|  | Core Elective | | | | Paper -3 | | | | | | | 3 | | | | | 3 | | **(to choose one out of 3)**   1. Software Testing 2. Software Project Management 3. Object Oriented Software | 25 | 75 | 100 |
| **External Major for other major Students (Inter/multi-disciplinary papers)** | | | | | | | | | | | | | | | | | | | | | | |
|  | Open Elective | | | Paper -3 | | | | | | | | 3 | | | | | 3 | | **(to choose one out of 3)**   1. Introduction to C 2. Introduction to C# 3. Introduction to Python | 25 | 75 | 100 |
|  | \*\*MOOC Courses | | |  | | | | | | | | - | | | | | - | |  | - | - | 100 |
|  |  | | |  | | | | | | | | **30** | | | | | **22** | |  | **200** | **600** | **900** |
| **SEMESTER 1V** | | | | | | | | | | | | | | | | | | |  | **CIA** | **Uni. Exam** | **Total** |
|  | Core | | | Paper-10 | | | | | | | | | 4 | | | | | 3 | Enterprise Java Programming | 25 | 75 | 100 |
|  | Core | | | Paper-11 | | | | | | | | | 4 | | | | | 3 | Python Programming | 25 | 75 | 100 |
|  | Core | | | Paper-12 | | | | | | | | | 4 | | | | | 3 | Web Applications using C# | 25 | 75 | 100 |
|  | Core | | | Project | | | | | | | | | 3 | | | | | 3 | Project Work (Compulsory) | 100  (75 Project + 25 viva) | | 100 |
|  | Practical | | | Paper-10 | | | | | | | | | 3 | | | | | 2 | Practical 10: Enterprise Java Programming | 25 | 75 | 100 |
|  | Practical | | | Paper-11 | | | | | | | | | 3 | | | | | 2 | Practical 11: Python Programming | 25 | 75 | 100 |
|  | Practical | | | Paper-12 | | | | | | | | | 3 | | | | | 2 | Practical 12:Web Applications using C# | 25 | 75 | 100 |
| **Internal Elective for same major students** | | | | | | | | | | | | | | | | | | | | | | |
|  | Core Elective | Paper -4 | | | | | | 3 | | | | | | 3 | | | | | **(to choose one out of 3)**   1. Internet of Things 2. Cloud Computing 3. Big Data Analysis | 25 | 75 | 100 |
| External Major for other major Students (Inter/multi-disciplinary papers) | | | | | | | | | | | | | | | | | | | | | | |
|  | Open Elective | Paper-4 | | | | | | 3 | | | | | | 3 | | | | | **(to choose one out of 3)**   1. Introduction to Database System 2. Introduction to IoT 3. Introduction to Mobile Application | 25 | 75 | 100 |
|  |  |  | | | | | | **30** | | | | | | **24** | | | | |  | **200** | **600** | **900** |
|  |  |  | | | | | | **120** | | | | | | **90** | | | | |  |  |  | **3600** |

**\* Field Study**

There will be field study which is compulsory in the first semester of all PG courses with 2 credits. This field study should be related to the subject concerned with social impact. Field and Topic should be registred by the students in the first semester of their study along with the name of a mentor before the end of the month of August. The report with problem identification and proposed solution should be written in not less than 25 pages in a standard format and it should be submitted at the end of second semester. The period for undergoing the field study is 30 hours beyond the instructional hours of the respective programme. Students shall consult their mentors within campus and experts outside the campus for selecting the field and topic of the field study. The following members may be nominated for confirming the topic and evaluating the field study report.

(i). Head of the respective department

(ii). Mentor

(iii). One faculty from other department

\*\***Mooc Courses**

Inclusion of the Massive Open Online Courses (MOOCs) with zero credits available on SWAYAM, NPTEL and other such portals approved by the University Authorities.

## SEMESTER III

## PAPER - 7

## ADVANCED JAVA PROGRAMMING

## COURSE OBJECTIVES

* To introduce programming with Applet and AWT.
* An overview of database access and details for managing information using the JDBC API.
* Examine the use of networking and collections.
* Learn how to program Servlet and JSP.
* To understand the web programming concepts in the perspective of Client and Server.

**UNIT -I: APPLETS AND GUI**

Applet Fundamentals- Applet Class - Applet lifecycle- Steps for Developing Applet Programs- Passing Values through Parameters- Graphics in Applets; GUI Application - Dialog Boxes - Creating Windows - Layout Managers – AWT Component classes – Swing component classes- Borders – Event handling with AWT components - AWT Graphics classes - File Choosers - Color Choosers – Tree – Table –Tabbed panels–Progressive bar - Sliders.

**UNIT- II: JDBC AND JAVA NETWORKING**

JDBC -Introduction - JDBC Architecture - JDBC Classes and Interfaces – Database Access with MySQL -Steps in Developing JDBC application - Creating a New Database and Table with JDBC - Working with Database Metadata; Java NetworkingBasics of Networking - Networking in Java- Socket Program using TCP/IP - Socket Program using UDP- URL and Inetaddressclasses.

**UNIT- III: COLLECTIONS AND DESIGN PATTERNS**

Collection Framework - ArrayList class - LinkedList class - ArrayListvs Linked List - ListIterator interface - HashSet class, LinkedHashSet class, TreeSet class PriorityQueue class - Map interface, HashMap class, LinkedHashMapclass ,TreeMap class - Comparable interface , Comparator interface, Comparable vs Comparator; Design Patterns: Introduction to Design patterns - Catalogue for Design Pattern - Factory Method Pattern, Prototype Pattern, Singleton Pattern, Adapter Pattern, Proxy Pattern, Decorator Pattern, Command Pattern, Template Pattern, Mediator Pattern;

**UNIT -IV: SERVLET AND JSP**

Servlet: Advantages over Applets - Servlet Alternatives - Servlet Strengths - Servlet Architecture - Servlet Life Cycle – GenericServlet, HttpServlet - First Servlet - Invoking Servlet - Passing Parameters to Servlets - Retrieving Parameters - Server-Side Include – Cookies; JSP : JSP Engines Working with JSP - JSP and Servlet - Anatomy of a JSP Page.

**UNIT -V: WEB PROGRAMMING**

Client-Side Programming: Client-side programming technologies - Form design using HTML, XHTML and DHTML and CSS - Client side validation Using JavaScript - Content Structuring using XML - Adding Interactivity with AJAX -JQuery Framework;

Server-side Programming: Web Servers - Handling request and response - Handling Form data - Session management - Database Access.

**TEXT BOOK**

1. S. Sagayaraj, R. Denis, P.Karthik& D. Gajalakshmi “Java Programming”, Universities Press, 2018.

**REFERENCES**

1. Patrick Naughton& Herbert Schildt, "The Complete Reference: Java 2", Tata McGraw Hill, 1999.
2. Deitel&Deitel, "Java How to Program", Prentice Hall, 5th Edition, 2002
3. Peter Haggar, "Practical Java: Programming Language Guide", Addison-

Wesley Pub Co, 1st Edition, 2000.

4. C.Muthu, ”Programming with Java”, McGraw Hill, Second Edition, 2008

**WEB REFERENCES**

http://math.hws.edu/javanotes/c6/index.html

http://www.tutorialspoint.com/awt/

[www.studytonight.com](http://www.studytonight.com)

[www.javatpoint.com](http://www.javatpoint.com)

[www.learnjavaonline.org](http://www.learnjavaonline.org)

[www.codingbat.com](http://www.codingbat.com)

**COURSE OUTCOMES**

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

* Develop Applet Programming using various techniques
* Develop applications using Abstract Window Toolkit and Events
* Update and retrieve the data from the databases using JDBC-ODBC
* Develop server side programs in the form of Servlets
* Build up Java Applications using collections and JSP Tags.

## PAPER - 8

## UNIX AND SHELL PROGRAMMING

**COURSE OBJECTIVES**

* To learn to add and remove users.
* To understand basic UNIX commands.
* To use controls structures.
* To understand loop structures.
* To understand System calls.

**UNIT – I : FILE ORGANIZATION**

Salient Features of Unix – Unix System Organization – Types of Shells – Unix Commands – The Unix File System – Creating Files – Listing Files and Directories. - The Boot Block – The Super Block – The Inode Table – Data Blocks – How Does Unix Access Files – Storage of Files – Disk Related Commands. System Administration: Adding and Removing Users – Daily Administration – Disk Management – Using a Raw Disk – Monitoring System Usage – Ensuring System Security – Providing Assistance to Users.

**UNIT - II: UNIX COMMANDS**

Password – Commands: cal, banner, touch – File Related Commands – Viewing Files – Taking Printouts – File Compression – I/O Redirection and Piping. vi Editor – Modes of operation – The First Editing Session. Processes in Unix: What’s Running Right Now – Still More Processes – Background Processes – The nohup command – Killing a process – Changing Process Priorities – Scheduling of Processes, Communication – Unix write and wall command - Basis of Unix Communication.

**UNIT - III: SHELL PROGRAMMING - I**

Interactive Shell Scripts – Shell Variables – Shell Keywords –Assigning Values to Variables – Positional Parameters – Passing Command Line Arguments – Setting Values of Positional Parameters – Displaying Date in Desired Format – Using Shift on Positional Parameters – Arithmetic in Shell Script- Taking Decisions.

**UNIT - IV SHELL PROGRAMMING - II**

Loop Control Structure: Loops – The While Loop – Reading from a file – The Until and for Loop – Creating Nested Directories – Generating Values for a for Loop – The Break and Continue Statement- Shell script using Command Line Arguments

**UNIT - V: SYSTEM CALLS**

System calls: File Structure related calls - create(), open(), close(), read(), write(), lseek(), process related calls- exec(), fork(), wait(), exit(), getpid(), getppid(), signal(), kill(), alarm() – Inter process communication calls– pipe().

**TEXT BOOK**

**Text**

1.Yashavant Kanetkar, “Unix Shell Programming”, BPB Publishers, New Delhi, 1996.

**Unit – I : Ch. 1, 2, 3, 15**

**Unit – II : Ch. 4, 5, 6, 7, 8**

**Unit – III : Ch. 9 - 10**

**Unit – IV : Ch. 11**

**Unit-V :** [**http://www.cs.utk.edu/~huangj/cs360/360/notes/Syscall**](http://www.cs.utk.edu/~huangj/cs360/360/notes/Syscall)

**Intro/lecture.html**

**REFERENCES**

1. Kernighan. et al. “The UNIX Programming Environment”, Second Edition, New Delhi: Prentice Hall of the India, 1988.
2. Stephen G. Kochan, Patrick Wood, “Unix Shell Programming”, Third Edition, Dorling Kindersley Pvt Ltd, Delhi, 2008.

**WEB REFERENCES**

**Online Tutorial**

1. http://www.cgl.ucsf.edu/Outreach/bmi219/slides/shell.html
2. http://www.cs.utk.edu/~huangj/cs360/360/notes/Syscall-Intro/lecture.html

**Online Quiz**

1. [www.tcyonline.com/tests/unix-and-shell-scripts](http://www.tcyonline.com/tests/unix-and-shell-scripts)

**Online Compiler**

1. www.compileonline.com/execute\_bash\_online.php*‎*

**COURSE OUTCOMES**

Upon successful completion of this course, the students should be able to:

* Learn to add and remove users.
* Understand basic UNIX commands.
* Use controls structures.
* Understand loop structures.
* Get familiarize with System calls concepts.

## PAPER - 9

## DESKTOP APPLICATION USING C#

## COURSE OBJECTIVES

* To know the differences between desktop and web application.
* To construct classes, methods, and accessor and instantiate objects.
* To create and manipulate GUI components in C#.
* To code solutions and compile C# projects within the .NET framework.
* To build own desktop application with Database

**UNIT - I: INTRODUCTION TO C#**

Introduction to .NET – Features of C# - Data Types – Value Types – Reference Types - Variables and Constants – Declaring – Assigning values – variables of nullable types – Operators – Type Conversions – Implicit and Explicit Type Conversions – Arrays – Single Dimensional and Multidimensional – Control Flow Statements – Selection – Iteration and Jump – Classes and Objects – Access Modifiers – Defining a Class – Variables – Properties and Methods – Creating Objects – Inheritance – Polymorphism- Constructor and Destructors.

**UNIT - II: WINDOWS FORMS**

Windows Forms – Form Class – Common Operations on Forms – Creating a Message Box –Handling Events – Mouse Events – Keyboard Events – Common Controls in Windows Forms – Label – TextBox – Button – Combo Box – List Box – Check Box – Radio Button – Group Box – Picture Box – Timer – Open File Dialog – Save File Dialog – Font Dialog – Color Dialog – Print Dialog – Tree View – Menu.

**UNIT - III: DELEGATES AND EVENTS**

Delegates – Declaring a Delegate – Defining Delegate Methods – Creating and Invoking Delegate Objects – Multicasting with Delegates – Events – Event Sources – Event Handlers – Events and Delegates.

**UNIT - IV: REFLECTION AND REMOTING**

Life Cycle of threads-Using Reflection – Reflecting the Members of a Class - Dynamic Loading and Reflection - .NET Remoting – Architecture – Hosting of Objects – Single Ton and Single Call – Remoting Server – Remoting Client.

**UNIT - V: DATABASE**

Creating Connection String – Creating a Connection to a Database – Creating a Command Object – Working with Data Adapters – Using Data Reader to work with Databases – Using Dataset.

**TEXT BOOKS**

1. Vikas Gupta , “Comdex .NET Programming “ , Dream Tech Press, New Delhi, 2011
2. Kogent Solutions, “ C# 2008 Programming Black Book”, Dream Tech Press, New Delhi, Platinum Edition, 2009

**REFERENCES**

1. Rebecca M.Riordon, “Microsoft ADO .Net 2.0 Step by Step”, Prentice Hall of India Private Limited, New Delhi, 2007
2. David S.Platt , “Introducing Microsoft .Net”, Prentice Hall of India( Private) Limited, Third Edition, New Delhi, 2006

**WEB REFERENCES**

http://csharp.net-tutorials.com/index.php

http://csharp.net-tutorials.com/classes/introduction/

http://www.homeandlearn.co.uk/csharp/csharp.html

http://www.indiabix.com/c-sharp-programming/questions-and-answers/

https://www.wiziq.com/online-tests/43860-c-basic-quiz

http://www.withoutbook.com/OnlineTestStart.php?quizId=71

http://www.compileonline.com/compile\_csharp\_online.php

http://www.ideone.com

**COURSE OUTCOMES**

After the completion of the course the students will be able:

* To know the differences between desktop application and web application.
* To construct classes, methods, and access modifier and instantiate objects.
* To create and manipulate GUI components in C# for windows application.
* To code solutions and compile C# projects within the .NET framework.
* To build the desktop application with Database.

## PRACTICAL-7

## ADVANCED JAVA PROGRAMMING

1. Develop Applet Programming with various techniques.
2. Develop applications using AWT.
3. Working with Graphics ,Color and Font
4. Working with JDBC Classes( Database Operations- Create, Insert, Delete, Update, Select)
5. Handling ResultSet and Statements.
6. Jasper Report Generation
7. Working with Servlet and JDBC
8. Handling Client/Server Networking
9. Develop Java Server Pages applications using JSP Tags.
10. Working with Java Collections.

## PRACTICAL - 8

## UNIX AND SHELL PROGRAMMING

Programming with Shell Script

1. Shell Script – sequential structure
2. Shell Script – Iterative control structure
3. Shell Script – Strings
4. Shell Script – Files
5. Shell Script – Command Line Arguments

System Calls

1. Printing the command line arguments
2. Read(), write(), open(), creat()
3. Execlp(), execvp(), perror(),
4. Use of fork(), wait() & exit()
5. Child process, generated interrupt &lseek()

## PRACTICAL - 9

## DESKTOP APPLICATIONS USING C#

1. Variables, Constants and Arrays
2. Classes and Objects
3. Inheritance
4. Polymorphism
5. Windows Form Controls (Label, Text, Button, Check Box, Radio)
6. Windows Form Controls (List, Combo, Timer, Group Box, Picture Box)
7. Menu Handling
8. Reflection
9. ADO.NET Connection
10. Data Command

## CORE ELECTIVE

## PAPER – 3

## (to choose one out of 3)

## A. SOFTWARE TESTING

**COURSE OBJECTIVES**

* To know the basic structure for testing teams.
* To expose the concept of test automation and test metrics.
* To know the different types of testing.

**UNIT – I: STRUCTURE FOR TESTING TEAMS AND TEST MANAGEMENT**

Dimensions of Organization Structures: Structures in Single-Product Companies-Structures for Multi-Product Companies- Effects of Globalization and Geographically Distributed Teams on Product Testing-Testing Services Organizations-Test Management: Test Planning-Test Process-Test Reporting.

**UNIT - II: SOFTWARE TEST AUTOMATION AND TEST METRICS**

Test Automation-Scope of Automation-Design and Architecture of Automation-Process Model for Automation-Selecting a Test Tool -Challenges in Automation-Test Metrics: Types of Metrics-Project Metrics-Progress Metrics-Productivity Metrics.

**UNIT – III: WHITE BOX, BLACK BOX AND INTEGRATION TESTING**

White Box Testing: Static Testing Structural Testing-Challenges in White Box Testing-Black Box Testing-Integration Testing: Types of Testing-Scenario Testing-Defect Bash.

**UNIT – IV: SYSTEM, PERFORMANCE AND REGRESSION TESTING**

System Testing: Functional System Testing-Non-Functional Testing-Acceptance Testing-Methodologies for Performance Testing –Tools for Performance Testing -Process for Performance Testing –Regression testing.

**UNIT – V: INTERNATIONALIZATION AND AD HOC TESTING**

Introduct**i**on-Primer on Internationalization-Enabling Testing-Local Testing-Language Testing-Localization Testing-Tools used for Internationalization-Ad hoc Testing: Pair Testing-Exploratory Testing-Iterative Testing-Agile and Extreme Testing. Software Testing Tools: WinRunner – Silk Test

**TEXT BOOK**

1. SrinivasanDesikan and Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson education, 2006. (latest edition)

**REFERENCES**

1. Boris Beizer, ”Software Testing Techniques” Second Edition, Dreamtech Press, New Delhi,2013.
2. K.V.KK. Prasad , Software Testing Tools, Dreamtech Press, New Delhi, 2005.
3. K.Mustafa and R.A.Khan, “Software Testing-Concepts and Practices”,Narosa Publishing House,New Delhi,2012.
4. William Perry, "Effective Methods for Software Testing", Wiley, New Delhi, 2009.
5. Mark C Paulk, Charles V Weber and Mary B Chrissis, "The Capability Maturity Model", Carnegie Mellon University, Pennsylvania, 2004.
6. John Watkins, “Agile Testing : How to succeed in an extreme Testing environment”, Cambridge Press, Cambridge, 2009

**WEB REFERENCES**

**Online Tutorial**

http://www.testingexcellence.com/istqb-quiz/

<http://withoutbook.com/OnlineTestStart.php?quizId=53>

<http://www.careerride.com/software-testing-quiz.aspx>

<http://en.wikipedia.org/wiki/HP_WinRunner>

## CORE ELECTIVE

## PAPER – 3

## B. SOFTWARE PROJECT MANAGEMENT

**COURSE OBJECTIVES**

* To provide sound knowledge in Project Management.
* To understand the importance of requirement gathering
* To explore different models in Software Development
* To know the workflow of a Project
* To identify various actors in the activity

**UNIT I: INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT**

Introduction:Project – Software Projects vs other types of Project – Activities Covered by SPM – Some Ways of Categorizing Software Projects – Stakeholders, Setting Objectives – The Business Case - Project Success and Failure - Management and Management Control. Project Evaluation:A Business Case – Project Portfolio Management – Evaluation of Individual Projects – Cost Benefit Evaluation – Risk Evaluation.

**UNIT II: PROJECT PLANNING AND SELECTION OF PROJECT APPROACH**

Project Planning - Introduction to Step Wise Project Planning – Step 0 to Step 10. Selection of an Appropriate Project Approach -Introduction – Build or Buy – Choosing Methodologies and Technologies – Software Processes and Process Models – Choice of Process Models – The Waterfall Model– Prototyping – other ways of categorizing prototype- Agile Methods – Extreme Programming - Selecting the Most Appropriate Process Model.

**UNIT III: EFFORT ESTIMATION AND ACTIVITY PLANNING**

Effort Estimation – Introduction –Estimates – Problems with Over and Under-estimate – Basis for Software Estimating – Effort Estimation Techniques – Bottom-up Estimating – Top-down Approach and Parametric Models – Expert Judgment - Estimating by Analogy – Albrecht Function Point Analysis – Function Mark II – COCOMO & COCOMO II – Cost Estimation – Staffing Pattern. Activity Planning –Introduction – Objectives of Activity Planning – When to plan – Project Schedules – Project and Activities – Sequencing and Scheduling Activities – Networking Planning Models – Formulating a Network Model– Activity on Arrow Networks.

**UNIT IV: RISK MANAGEMENT, RESOURCE ALLOCATION AND MONITORING**

Risk Management –Risk – Categories of Risk – A Framework for Dealing with Risk – Risk Identification – Risk Assessment – Risk Planning – Risk Management. Resource Allocation –Introduction – The Nature of Resources – Identifying Resource Requirements – Scheduling Resources. Monitoring –Creating the Framework – Collecting the Data – Review and Project Termination Review – Visualizing Progress – Cost Monitoring and Earned Value Analysis – Getting the Project Back to Target – Change Control – SCM.

**UNIT V: MANAGING PEOPLE AND WORKING IN TEAMS**

Managing People –Understanding Behavior – Organizational Behavior – Selecting the Right Person for the Job – Instruction in the Best Methods – Motivation – The Oldham-Hackman Job Characteristics Model – Stress – Health and Safety. Working in Teams –Introduction – Becoming a Team – Decision Making – Organization and Team Structures – Coordination Dependencies – Dispersed and Virtual Teams – Communication Genres – Communication Plans – Leadership.

**TEXT BOOK**

1. BOB Huges, Mike Cotterell, Rajib Mall “Software Project Management”, McGraw Hill, Fifth Edition,2011.

**REFERENCES**

1. Futrell, “Quality software Project management”, Pearson Education India.
2. Royce, “Software Project Management”, Pearson Education India.

**WEB REFERENCES**

<https://www.lynda.com/Project-Management-training-tutorials/39-0.html>

[www.rspa.com/spi/project-mgmt.html](http://www.rspa.com/spi/project-mgmt.html)

**COURSE OUTCOMES**

Upon completion of the course students will be able to:

* Understand the activities during the project scheduling of any software application.
* Learn the risk management activities and the resource allocation for the projects.
* Apply the software estimation and recent quality standards for evaluation of the software Projects.
* Acquire knowledge and skills needed for the construction of highly reliable software project.
* Able to create reliable, replicable cost estimation that links to the requirements of project planning and managing.

## CORE ELECTIVE

## PAPER - 3

## C. OBJECT ORIENTED SOFTWARE

**COURSE OBJECTIVES**

**Unit – I:**

Introduction to objects - module - cohesion - coupling - data encapsulation - abstract data types - information hiding - objects, - inheritance - polymorphism & dynamic binding - cohesion & coupling of objects. Reusability, protability& interoperability - reuse concepts - impediments to reuse, reuse case studies - objects & productivity - reuse during design & implementation phases - reuse & maintenance, portability, why portability, techniques for achieving portability - ihnteroperability - future trends in interoperability.

**Unit – II:**

Planning and estimation - planning and the software process - estimating duration and cost - components of a software project management plan - software project management plan frame work - IEEE software project management plan - planning of testing - planning of object oriented projects - training requirements - documentation standards - CASE tools for planning and estimating - testing the software project management requirements phase - requirements analysis techniques - reusing the prototyping - human factors - rapid prototyping as a specification technique - reusing the rapid prototyping - other uses of rapid prototyping - management implication of the application design (JAD) - Comparison of requirement analysis techniques - testing during requirement phase - CASE tools for the requirement phase - metrics for the requirement phase - obsertoglesby case study: requirements phase obsertoglesby case study - rapid prototype - object oriented requirements.

**Unit – III:**

Specification phase - specification document informal - specification - structured, systems analysis - other semi-formal techniques - entity relationship modeling - finite. state machines - Petrinets z357 - other formal techniques - comparison of specification techniques - testing during specification phase - CASE tools for the specification phase - metrics for the specification phase - obvert oglesy case study: Structured systems analysis - software project management. Object oriented analysis phase - object oriented versus structured paradigm - object oriented analysis - elevator problem - use case modeling - dynamic modeling - testing during object oriented analysis phase - case tools - software project management.

**Unit – IV:**

Design phase - design and abstraction - action oriented design - data flow analysis - transaction analysis - data oriented design - object oriented design - elevator problem - formal techniques for detail designs - real time design techniques - testing - case tools - metrics - object oriented design.- Implementation phase: choice or programming language - forth generation language - good programming practice - coding standards - module reuse - module test case selection - black box - glass box module testing techniques - comparison clean room - potential problems when testing objects - management aspects of module testing- CASE tools for implementation phase.

**Unit – V:**

Implementation and integration phase - testing - graphical user interfaces - product testing - acceptance testing - case tools for this phase - integration environments for business applications - public tools infrastructure - potential problems with environments. Maintenance phase - why maintenance is necessary - case study - management - maintenance of object oriented software - maintenance skill versus development skills - reverse engineering - testing - case tools.

**TEXT BOOK**

* 1. Stephen R. Schach - Classical and Object oriented Software Engineering 4th Edition -

McGraw Hill

2. Ivar Jacobson - Object Oriented Software Engineering - Addison Wesley.

**REFERENCES**

1. Grady Booch, Object Oriented Analysis and Design, Addison-Wesley. 5 ed 2009

2. Grady Booch, James Rumbaugh and Ivar Jacobson, Unified Modeling Language Guide,

Addison-Wesley. 5ed 2009

3. Erich Gamma et al., Design Patterns: Elements of Reusable OO Software, Addison-

Wesley.5 ed 2010

4. Michael L. Scott, Programming Language Pragmatics, Morgan-Kaufmann.5 ed 2006

5. Kim Bruce, Foundations of Object Oriented Languages, Prentice-Hall.6 2011

**WEB REFERENCES**

http:// https://www.tutorialspoint.com.objecrroreientedsoftwareengineering/>

<http://www.indiabix.com/online-test/objectorientedsoftwareengineeringtest/

**COURSE OUTCOMES**

On successful completion of course, learners will be able to:

* Understand and demonstrate basic knowledge in object oriented software
* Identify requirements, analyze and prepare models.
* Plan, schedule and track the progress of the projects.
* Design & develop the software projects
* Identify risks, manage the change to assure quality in software projects.
* Apply testing principles on software project and understand the maintenance concepts.

## OPEN ELECTIVE

## PAPER - 3

## (to choose one out of 3)

## A. INTRODUCTION TO C

**COURSE OBJECTIVES**

* To identify​situations where computational methods and computers would be useful.
* To enhance their analyzing and problem-solving skills and use the same for writing programs in C.
* To develop logics and that will help them to create programs, applications in C.
* To identify programming task involvedin a given computational problem.
* To approach​the programming tasks using techniques learned and write​pseudo-code.
* To choose​the right data representation formats based on the requirements of the problem.
* To use​the comparisons and limitations of the various programming constructs and choose​the right one for the task in hand.
* To enter the program on a computer, edit, compile, debug, correct, recompile and run it.
* To identify tasks in which the numerical techniques learned are applicable and apply them to write programs.

**UNIT – I: DATA TYPES, OPERATORS AND STRUCTURES**

Structure of a C program – Basic data types (int, float, char, double, void) – constants and variables (variable declaration, integer, real,float, character, variables) – operators and expressions (arithmetic operators, relational operators, logical operators, bitwise operators, type casting, type conversion, enumerated data type, typedef) – Control Constructs (if, switch, while, do…while, for, break and continue, exit() function, goto and label).

**UNIT – II: ARRAYS AND FUNCTIONS**

Arrays (declaration, one and two dimensional arrays) - Character Arrays and Strings. Function Fundamentals (General form, Function Definition, Function arguments, return value) – Parameter passing: call-by-value and call-by-reference – Recursion – Passing Arrays to Function – Passing Strings to Function.

**UNIT – III: POINTERS**

Understanding Pointers – Accessing the Address of a Variable – Declaring the Pointer Variables – Initialization of Pointer Variables – Accessing a Variable through its Pointer – Pointer Expressions – Pointers and Arrays – Pointers and Character Strings – Array of Pointers – Pointers as Function Arguments – Functions returning Pointers – Pointers to Functions.

**UNIT – IV: STORAGE CLASSES, STRUCTURES AND UNIONS**

Scope rules (Local variables and global variables, scope rules of functions) -Type modifiers and storage class specifier.

Structures – Basics of Structure – Declaring of Structure – Referencing Structure elements - Array of Structures – Nesting of Structures - Passing Structures to function – Pointers and Structures - Unions.

**UNIT – V: FILE MANAGEMENT IN C**

Introduction – Defining and Opening a File – Closing a File – Input / Output Operations on Files – Command Line Arguments.

**TEXT BOOK**

1. E.Balagurusamy, “Programming in ANSI C”, Seventh Edition, McGraw Hill Education Private Limited, NewDelhi: 2017.

**REFERENCES**

1. YashavantKanetkar, “Let us C”, BPB Publications, Tenth Edition - New Delhi: 2010

2. Ashok N.Kamthane, “Programming in C”, Second Impression, Pearson: 2012.

**WEB REFERENCES**

<http://www.c4learn.com/?gclid=COK1y6nHk7wCFcUA4g>odmlgAKA/

<http://www.cprogramming.com/tutorial/c-tutorial.html/>

<http://www.tutorialspoint.com/cprogramming/>

**COURSE OUTCOMES**

After course completion the students will have the following learning outcomes:

* Understanding a functional hierarchical code organization.
* Ability to define and manage data structures based on problem subject domain.
* Ability to work with textual information, characters and strings.
* Ability to work with arrays, structures, pointers and files.

## OPEN ELECTIVE

## PAPER - 3

## B. INTRODUCTION TO C#

## COURSE OBJECTIVES

* To know the differences between desktop and web application.
* To construct classes, methods, and accessor and instantiate objects.
* To create and manipulate GUI components in C#.
* To code solutions and compile C# projects within the .NET framework.
* To build own desktop application with Database

**UNIT - I: INTRODUCTION TO C#**

Introduction to .NET – Features of C# - Data Types – Value Types – Reference Types - Variables and Constants – Declaring – Assigning values – variables of nullable types – Operators – Type Conversions – Implicit and Explicit Type Conversions – Arrays – Single Dimensional and Multidimensional – Control Flow Statements – Selection – Iteration and Jump – Classes and Objects – Access Modifiers – Defining a Class – Variables – Properties and Methods – Creating Objects – Inheritance – Polymorphism- Constructor and Destructors.

**UNIT - II: WINDOWS FORMS**

Windows Forms – Form Class – Common Operations on Forms – Creating a Message Box –Handling Events – Mouse Events – Keyboard Events – Common Controls in Windows Forms – Label – TextBox – Button – Combo Box – List Box – Check Box – Radio Button – Group Box – Picture Box – Timer – Open File Dialog – Save File Dialog – Font Dialog – Color Dialog – Print Dialog – Tree View – Menu.

**UNIT - III: DELEGATES AND EVENTS**

Delegates – Declaring a Delegate – Defining Delegate Methods – Creating and Invoking Delegate Objects – Multicasting with Delegates – Events – Event Sources – Event Handlers – Events and Delegates.

**UNIT - IV: REFLECTION AND REMOTING**

Life Cycle of threads-Using Reflection – Reflecting the Members of a Class - Dynamic Loading and Reflection - .NET Remoting – Architecture – Hosting of Objects – Single Ton and Single Call – Remoting Server – Remoting Client.

**UNIT - V: DATABASE**

Creating Connection String – Creating a Connection to a Database – Creating a Command Object – Working with Data Adapters – Using Data Reader to work with Databases – Using Dataset.

**TEXT BOOKS**

1. Vikas Gupta , “Comdex .NET Programming “ , Dream Tech Press, New Delhi, 2011
2. Kogent Solutions, “ C# 2008 Programming Black Book”, Dream Tech Press, New Delhi, Platinum Edition, 2009

**REFERENCES**

1. Rebecca M.Riordon, “Microsoft ADO .Net 2.0 Step by Step”, Prentice Hall of India Private Limited, New Delhi, 2007
2. David S.Platt , “Introducing Microsoft .Net”, Prentice Hall of India( Private) Limited, Third Edition, New Delhi, 2006

**WEB REFERENCES**

http://csharp.net-tutorials.com/index.php

http://csharp.net-tutorials.com/classes/introduction/

http://www.homeandlearn.co.uk/csharp/csharp.html

http://www.indiabix.com/c-sharp-programming/questions-and-answers/

https://www.wiziq.com/online-tests/43860-c-basic-quiz

http://www.withoutbook.com/OnlineTestStart.php?quizId=71

http://www.compileonline.com/compile\_csharp\_online.php

http://www.ideone.com

**COURSE OUTCOMES**

After the completion of the course the students will be able:

* To know the differences between desktop application and web application.
* To construct classes, methods, and access modifier and instantiate objects.
* To create and manipulate GUI components in C# for windows application.
* To code solutions and compile C# projects within the .NET framework.
* To build the desktop application with Database.

## OPEN ELECTIVE

## PAPER - 3

## C. INTRODUCTION TO PYTHON

**COURSE OBJECTIVES**

* To know the basics of algorithmic problem solving
* To read and write simple Python programs.
* To develop Python programs with conditionals and loops.
* To define Python functions and call them.
* To use Python data structures – lists, tuples, dictionaries.
* To do input/output with files in Python.

**UNIT - I: OVERVIEW**

Introduction to Python: Features of Python - How to Run Python – Identifiers - Reserved Keywords - Variables - Comments in Python - Indentation in Python - Multi-Line Statements - Multiple Statement Group (Suite) – Quotes in Python - Input, Output and Import Functions - Operators. Data Types and Operations: Numbers-Strings-List-Tuple-Set-Dictionary-Data type conversion.

**UNIT - II: FLOW CONTROL & FUNCTIONS**

Flow Control: Decision Making-Loops-Nested Loops-Types of Loops. Functions: Function Definition-Function Calling - Function Arguments - Recursive Functions - Function with more than one return value.

**UNIT - III: MODULES, PACKAGES AND FILE HANDLING**

Modules and Packages: Built-in Modules - Creating Modules - import Statement - Locating Modules - Namespaces and Scope - The dir() function - The reload() function - Packages in Python - Date and Time Modules. File Handling: Opening a File - Closing a File - Writing to a File – Reading from a File - File Methods - Renaming a File - Deleting a File - Directories in Python.

**UNIT - IV: OBJECT ORIENTED PROGRAMMING**

Class Definition - Creating Objects - Built-in Attribute Methods - Built-in Class Attributes - Destructors in Python Encapsulation - Data Hiding- Inheritance - Method Overriding Polymorphism. Exception Handling: Built-in Exceptions - Handling Exceptions - Exception with Arguments- Raising Exception - User-defined Exception - Assertions in Python

**UNIT - V: REGULAR EXPRESSIONS & WEB APPLICATIONS**

Regular Expressions: The match() function - The search() function - Search and Replace - Regular Expression Modifiers: Option Flags - Regular Expression Patterns - Character Classes - Special Character Classes - Repetition Cases - findall() method - compile() method. Web Application Framework- Django Architecture- Starting development- Case Study: Blogging App.

**TEXT BOOKS**

1. Jeeva Jose and P. SojanLal, “Introduction to Computing and Problem Solving with Python”, Khanna Book Publising Co. (P) Ltd., 2016.

2. ArshdeepBahga, Vijay Madisetti, “Cloud Computing: A Hands – On Approach” Universities press (India) Pvt. limited 2016.

**REFERENCES**

1. Wesley J. Chun, “Core Python Programming”, Second Edition, Prentice Hall Publication, 2006.

2. Timothy A Budd, “Exploring Python”, Tata McGraw Hill, New Delhi, ISBN: 780071321228

**WEB REFERENCES**

www.learnpython.org/

<https://www.codecademy.com/learn/python>

https://www.Codementor.io

<https://www.Python.org>

**COURSE OUTCOMES**

Upon completion of the course, students will be able:

* To explore the fundamental concepts of Python
* To understand Basics of Python programming language
* To solve simple problems using Python
* To acquire fundamental knowledge and skills on Python Programming
* To understand the nuances of this language.
* To know the usage of modules and packages in Python
* To familiarize with file concepts in Python
* To familiarize with web concepts using Python.

**SEMESTER IV**

## PAPER - 10

## ENTERPRISE JAVA PROGRAMMING

**COURSE OBJECTIVES**

* To expose the complete knowledge of MVC, Java server faces and Enterprise java bean.
* Develop Enterprise web application using EJB.

**UNIT – I:Controlling Web Application Behavior with web.xml**

Understanding the purpose of web.xml - Customizing URLs - Turning off default URLs -Initializing servlets and JSP pages - Preloading servlets and JSP pages - Declaring filters - Designating welcome pages and error pages - Restricting access to Web resources.

**UNIT – II: JAVA SERVER FACES *(*JSF)**

JSP Benefits, Framework roles, Simple JSF application, User Interface Component Model, Navigational Model, Life Cycle of JSF page, Using JSF in JSP Pages – Setting up a page, using core tags, using HTML tags, using localized messages, Using converters.

**UNIT – III:DEVELOPING WITH JAVASERVER FACES TECHNOLOGY**

Registering listeners on components, validators, binding component values to external data sources, referencing a backing Bean method, using custom objects writing component properties, performing localization, creating custom converter, implementing event listener, creating custom validator, writing backing Bean methods.

**UNIT – IV: ENTERPRISE JAVA BEANS (EJB)**

Introduction to Enterprise Beans, Session Bean, Entity Bean, Message driven Bean, defining clients access with interfaces, contents of an enterprise Bean, life cycle of enterprise Bean, creation of Enterprise Bean, application client, web client, other Enterprise Bean features, handling exceptions, mapping table relationships for Bean managed persistence, primary keys for bean managed persistence, container managed persistence – primary key for container managed persistence, Message driven Bean example – applicant client and message driven Bean class.

**UNIT – V: JAVA MESSAGE SERVICE API**

Basic JMS API concepts – architecture, messaging domains – consumption- JMS API programming model – administered objects - connections - sessions - message producers- consumers – messages - exception handling - JMS client applications - creating robust JMS applications - Using JMS API in J2EE application.

**TEXT BOOK**

**Unit I**

1. Marty Hall, Larry Brown, YaakovChaikin “Core Servlets and JavaServer Pages, Volume 2:advanced Technologies”, 2nd Edition, Prentice Hall PTR.

**Unit II, III, IV & V**

1. Stephanie Bodoffetl., “The J2EETM Tutorial”, Pearson Education, 2005.

**REFERENCES**

1. David Geary, Cay Horstmann ,”Core JavaServer™ Faces”,2nd Edition, Prentice Hall,2007.
2. Bill Dudney, Jonathan Lehr, Bill Willis, LeRoy Mattingly, “Mastering JavaServer Faces”,Wiley publications.
3. Debu Panda, Reza Rahman, Derek Lane,”EJB 3 in Action”, Manning publications.

**WEB REFERENCES**

**Online Tutorial**

[www.corejsf.com](http://www.corejsf.com)

[www.roseindia.net](http://www.roseindia.net)

[www.r4r.co.in](http://www.r4r.co.in)

[www.mkyong.com](http://www.mkyong.com)

[www.java2s.com](http://www.java2s.com).

**COURSE OUTCOMES**

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

* Develop JSF page using various techniques
* Develop applications using Java Beans
* Working with Java API’s for creating applications.

## PAPER - 11

## PYTHON PROGRAMMING

**COURSE OBJECTIVES**

* To know the basics of algorithmic problem solving
* To read and write simple Python programs.
* To develop Python programs with conditionals and loops.
* To define Python functions and call them.
* To use Python data structures – lists, tuples, dictionaries.
* To do input/output with files in Python.

**UNIT - I: OVERVIEW**

Introduction to Python: Features of Python - How to Run Python – Identifiers - Reserved Keywords - Variables - Comments in Python - Indentation in Python - Multi-Line Statements - Multiple Statement Group (Suite) – Quotes in Python - Input, Output and Import Functions - Operators. Data Types and Operations: Numbers-Strings-List-Tuple-Set-Dictionary-Data type conversion.

**UNIT - II: FLOW CONTROL & FUNCTIONS**

Flow Control: Decision Making-Loops-Nested Loops-Types of Loops. Functions: Function Definition-Function Calling - Function Arguments - Recursive Functions - Function with more than one return value.

**UNIT - III: MODULES, PACKAGES AND FILE HANDLING**

Modules and Packages: Built-in Modules - Creating Modules - import Statement - Locating Modules - Namespaces and Scope - The dir() function - The reload() function - Packages in Python - Date and Time Modules. File Handling: Opening a File - Closing a File - Writing to a File – Reading from a File - File Methods - Renaming a File - Deleting a File - Directories in Python.

**UNIT - IV: OBJECT ORIENTED PROGRAMMING**

Class Definition - Creating Objects - Built-in Attribute Methods - Built-in Class Attributes - Destructors in Python Encapsulation - Data Hiding- Inheritance - Method Overriding Polymorphism. Exception Handling: Built-in Exceptions - Handling Exceptions - Exception with Arguments- Raising Exception - User-defined Exception - Assertions in Python

**UNIT - V: REGULAR EXPRESSIONS & WEB APPLICATIONS**

Regular Expressions: The match() function - The search() function - Search and Replace - Regular Expression Modifiers: Option Flags - Regular Expression Patterns - Character Classes - Special Character Classes - Repetition Cases - findall() method - compile() method. Web Application Framework- Django Architecture- Starting development- Case Study: Blogging App.

**TEXT BOOKS**

1. Jeeva Jose and P. SojanLal, “Introduction to Computing and Problem Solving with Python”, Khanna Book Publising Co. (P) Ltd., 2016.

2. ArshdeepBahga, Vijay Madisetti, “Cloud Computing: A Hands – On Approach” Universities press (India) Pvt. limited 2016.

**REFERENCES**

1. Wesley J. Chun, “Core Python Programming”, Second Edition, Prentice Hall Publication, 2006.

2. Timothy A Budd, “Exploring Python”, Tata McGraw Hill, New Delhi, ISBN: 780071321228

**WEB REFERENCES**

www.learnpython.org/

<https://www.codecademy.com/learn/python>

https://www.Codementor.io

<https://www.Python.org>

**COURSE OUTCOMES**

Upon completion of the course, students will be able:

* To explore the fundamental concepts of Python
* To understand Basics of Python programming language
* To solve simple problems using Python
* To acquire fundamental knowledge and skills on Python Programming
* To understand the nuances of this language.
* To know the usage of modules and packages in Python
* To familiarize with file concepts in Python
* To familiarize with web concepts using Python.

## PAPER - 12

## WEB APPLICATION USING C#

**COURSE OBJECTIVES**

* To know the differences between desktop and web application.
* To construct classes, methods, and accessor and instantiate objects.
* To create and manipulate GUI components in C#.
* To code solutions and compile C# projects within the .NET framework.
* To build own desktop application with Database

**UNIT- I: INTRODUCTION TO ASP.NET AND WEB FORMS**

Developing ASP.NET Applications - ASP.NET File Types - The bin Directory - Application Updates - A Simple Application from Start to Finish-web.config file Web Form Fundamentals - A Simple Page Applet - The Problem With Response.Write - Server Controls - HTML Server Controls - ViewState - The HTML Control Classes - Events - Event Handling Changes - The Currency Converter application-Adding Support for Multiple Currencies - Adding Linked Images - Setting Styles – A Deeper Look at HTML control classes-HTML control events-The HTML control Base class-The HtmContainerControl Class-The HtmlInputControl Class-The Page class-The Controls collection-The HttpRequest Class-The HttpResponse Class-The ServerUtility Class-Assessing HTML Server controls

**UNIT - II: WEB CONTROLS**

Web Controls - Stepping Up to web Controls - Basic Web Control Classes - The web Control Tags - The WebControl Base Class - Units Enumerated Values - Colors - Fonts - List Controls - Table Controls - AutoPostBack and Web Control Events - How Postback Events Work - The Page Lifecycle - The Greeting Card Applet - Validation and rich Controls- The Calendar Control-Formatting the Calendar-restricting Dates- The AdRotator control-The Wizard control-Validation-The Validation Controls -The Validation Process-The Validator Class-A Simple Validation Example –Sever side example-Manual Validation-Understanding Regular Expressions-Literals and MetaCharacters-Finding a Regular expression- A Validated Customer Form

**UNIT - III: COMPONENT BASED PROGRAMMING**

Introduction – Creating a Simple Component – Properties and State – Database Components – Consuming the Database Component – Enhancing the Component with Error Handling – Aggregate Information – Data Objects.

**UNIT - IV: CUSTOM CONTROLS**

User Controls – Creating a Simple User Control – Visual Studio.NET Custom Control Support – Independent User Controls – Integrated User Controls – User Control Events – Limitations – Deriving Custom Controls.

**UNIT - V: DATABASE ACCESS WITH COMMAND, ADAPTER AND XML**

ADO.NET Data Access - About the ADO.NET Example - Obtaining the Sample Database -Simple Data Access - Simple Data Update - Importing the Namespaces - Creating a Connection - The Connection String SQL - Making the Connection - Defining the Select Command - Using a Command with a DataReader - Updating Data - Using Update - Insert - and Delete Commands - Accessing Disconnected Data - Selecting Disconnected Data - Selecting Multiple Tables - Modifying Disconnected Data - Modifying and Deleting Rows - Adding Information - to a DataSet - Updating Disconnected Data - The Command Builder - Updating a DataTable - Controlling Updates - An Update Example – Using XML - XML’s Hidden Role in .NET - XML Basics - Attributes - Comments - The XML Classes - the XML TextWriter - The XML Text Reader - Working with XML Documents - Reading an XML Document - Searching an XML Document - XML Validation – CreatingXML Schema -XSD Documents - Validating an XML File.

**TEXT BOOKS**

1. Mathew MacDonald, “ASP.NET: The Complete Reference”, Tata McGraw Hill Publishing Company Ltd., New Delhi, 2006
2. Dino Eesposito, “Introducing Microsoft ASP.NET 2.0”, AsokeK.Ghosh, Prentice Hall of India, Eastern Economy Edition, New Delhi, 2006

**REFERENCE**

1. Stephen Walther,”ASP.NET 3.5 Unleashed“, Pearson Education, Dorling Kindersley Pvt. Ltd, Second Edition, 2008

**WEB REFERENCES**

1. http://csharp.net-tutorials.com/index.php
2. http://csharp.net-tutorials.com/classes/introduction/
3. http://www.homeandlearn.co.uk/csharp/csharp.html
4. http://www.indiabix.com/c-sharp-programming/questions-and-answers/
5. https://www.wiziq.com/online-tests/43860-c-basic-quiz
6. http://www.withoutbook.com/OnlineTestStart.php?quizId=71
7. http://www.compileonline.com/compile\_csharp\_online.php
8. <http://www.ideone.com>

**COURSE OUTCOMES**

After the completion of the course the students will be able:

* To know the differences between desktop application and web application.
* To construct classes, methods, and access modifier and instantiate objects.
* To create and manipulate GUI components in C# for windows application.
* To code solutions and compile C# projects within the .NET framework.
* To build the desktop application with Database.

## PRACTICAL-10

## ENTERPRISE JAVA PROGRAMMING

1. Simple JSP Application
2. JSF in JSP Pages
3. Using all HTML render kit
4. Using all Core render kit
5. Creating Enterprise Bean
6. Creating Web Client
7. Using Session Bean
8. Bean Managed Persistence and Container Managed Persistence
9. Creating Simple JMS Client Applications
10. Creating Robust JMS Applications

## PRACTICAL-11

## PYTHON PROGRAMMING

1. Working with numbers
2. Implementing String operations
3. Working with Tuples and Set
4. Implementation of Dictionaries
5. Demonstrating List Operations.
6. Flow Control and Functions
7. Modules and Packages
8. File handling
9. Object Oriented Programming
10. Exception Handling and Regular Expressions

## PRACTICAL-12

## WEB APPLICATIONS USING C#

1. Web Configuration File
2. Viewstate
3. HTML Control Classes, Control Events, Container and Input Control Classes,
4. Web Control Classes & Control Tags
5. Validation Controls
6. Rich Controls
7. Data Access
8. Components
9. Custom Controls
10. User Controls

## CORE ELECTIVE

## PAPER - 4

## (to choose one out of 3)

## A. INTERNET OF THINGS

**OBJECTIVES**

## To design and Develop IOT based solution for real world applications

## To realize the evolution of Internet in Mobile Devices, Cloud & Sensor Networks

## To understand the building blocks of Internet of Things and its characteristics.

## To understand the concepts of IOT and its application.

**UNIT - I: INTRODUCTION**

Introduction and Definition of Internet of Things, IoT Growth – A Statistical View, Application Areas of IoT, Characteristics of IoT, Things in IoT, IoT Stack, Enabling Technologies, IoT Challenges, IoT Levels, Is Cyber Physical System the same as IoT? Is WSN the same as IoT?

**UNIT - II: INTRODUCTION TO SENSORS, MICROCONTROLLERS, AND THEIR INTERFACING**

Introduction to Sensor Interfacing, Types of Sensors, Controlling Sensors through Webpages, Microcontrollers: A Quick Walkthrough, ARM. Protocols for IoT – Messaging and Transport Protocols, Messaging Protocols (MQTT, CoAP, AMQP), Transport Protocols (Li-Fi, BLE).

**UNIT - III: PROTOCOLS FOR IOT**

Addressing and Identification, Internet Protocol Version 4 (IPv4), Internet Protocol Version 6 (IPv6), Uniform Resource Identifier (URI). Cloud for IoT - Introduction, IoT with Cloud – Challenges, Selection of Cloud Service Provider for IoT Applications: An Overview, Introduction to Fog Computing, Cloud Computing: Security Aspects, Case Study: How to use Adafruit Cloud? Application of Data Analytics in IOT.

**UNIT - IV: APPLICATION BUILDING WITH IOT**

Introduction, Smart Perishable Tracking with IoT and Sensors, Smart Healthcare – Elderly Fall Detection with IoT and Sensors, Smart Inflight Lavatory Maintenance with IoT, IoT–Based Application to Monitor Water Quality, Smart Warehouse Monitoring – Let the Drone Fly for You, Smart Retail – IoT Possibilities in the Retail Sector, Prevention of Drowsiness of Drivers by IoT-Based Smart Driver Assistance Systems, System to Measure Collision Impact in an Accident with IoT.

**UNIT - V: GETTING FAMILIARIZED WITH ARDUINO IDE**

Architecture, Arduino Programming, A Simple Application, Arduino Playground. Getting Familiarized with Raspberry Pi - Story behind Raspberry Pi, Architecture, Compatible Peripherals, Add-Ons, and Accessories, Operating System for Raspberry Pi, Setting up Raspberry Pi, Initial Configuration for Raspberry Pi, Linux Based Softwares in Raspberry Pi, Application Development with Raspberry-Pi – A Quick Walk Through.

**TEXT BOOK**

1. Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, Internet of Things, Wiley, India, 2019.

**REFERENCES**

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on Approach)”, 1stEdition, VPT, 2014.
2. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013.

**WEB REFERENCES**

1. https://www.coursera.org/courses?query=iot
2. https://online.stanford.edu/courses/xee100-introduction-internet-things

## https://www.tutorialspoint.com/internet\_of\_things/index.htm

## OUTCOMES

## By the end of the course, the student shall be able to

## Design and Develop IOT based solution for real world applications

## Realize the evolution of Internet in Mobile Devices, Cloud & Sensor Networks

## Understand building blocks of Internet of Things and its characteristics.

## Understand the concept of IOT and its application.

## CORE ELECTIVE

## PAPER - 4

## B. CLOUD COMPUTING

**COURSE OBJECTIVES**

* To introduce the broad perceptive of cloud architecture and model.
* To understand the concept of parallel and distributed computing
* To be familiar with the different technologies.
* To understand the features of virtualization.
* To learn to design the trusted cloud Computing system with different cloud platforms

**UNIT - I: INTRODUCTION**

Cloud Computing at a Glance, The Vision of Cloud Computing, Deﬁning a Cloud, Cloud Computing Reference Model, Characteristics and Beneﬁts, Challenges Ahead, Historical Developments - Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments - Application Development, Infrastructure and System Development, Computing Platforms and Technologies - Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com

**UNIT – II: PRINCIPLES OF PARALLEL AND DISTRIBUTED COMPUTING**

Parallel vs. Distributed Computing , Elements of Parallel Computing - Hardware Architectures for Parallel Processing, Approaches to Parallel Programming, Levels of Parallelism, Laws of Caution, Elements of Distributed Computing - General Concepts and Deﬁnitions, Components of a Distributed System, Architectural Styles for Distributed Computing, Models for Inter-Process Communication, Technologies for Distributed Computing - Remote Procedure Call, Distributed Object Frameworks, Service Oriented Computing.

Virtualization - Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques, Execution Virtualization, and Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples - Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V

**UNIT - III: CLOUD COMPUTING ARCHITECTURE**

Introduction, Cloud Reference Model - Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds - Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges - Cloud Deﬁnition, Cloud Interoperability and Standards, Scalability and Fault Tolerance, Security, Trust, and Privacy, Organizational Aspects. High-Throughput Computing: Task Programming - Task Computing, Characterizing a Task, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Aneka Task-Based Programming.

**UNIT - IV: ANEKA**

Cloud Application Platform - Framework Overview, Anatomy of the Aneka Container - From the Ground Up: Platform Abstraction Layer, Fabric Services, Foundation Services, Application Services, Building Aneka Clouds - Infrastructure Organization Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management - Aneka SDK , Management Tools. Concurrent Computing: Thread Programming- Introducing Parallelism for Single Machine Computation, Programming Applications with Threads - Techniques for Parallel Computation with Threads, Multithreading with Aneka - Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads - Aneka Threads Application Model, Domain Decomposition: Matrix MultiplicationFunctional Decomposition: Sine, Cosine, and Tangent.

**UNIT - V: CLOUD PLATFORMS IN INDUSTRY**

Amazon Web Services - Compute Services, Storage Services, Communication Services, Google AppEngine - Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure - Azure Core Concepts - SQL Azure - Windows Azure Platform Appliance. Cloud Applications - Scientiﬁc Applications - Healthcare: ECG Analysis in the Cloud - Biology: Protein Structure Prediction - Biology: Gene Expression Data Analysis for Cancer Diagnosis - Geoscience: Satellite Image Processing, Business and Consumer Applications - CRM and ERP - Productivity - Social Networking - Media Applications - Multiplayer Online Gaming. Advanced Topics in Cloud Computing - Energy Efﬁciency in Clouds, Market Based Management of Clouds, Federated Clouds / InterCloud, Third Party Cloud Services

**TEXT BOOKS**

1**.** Rajkumar Buyya, Christian Vecchiola, and S. ThamaraiSelvi. Mastering cloud computing: foundations and applications programming. Tata McGraw Hill Education Private Limited, New Delhi , 2013

**REFERENCES**

1. Rittinghouse and Ransome, Cloud Computing: Implementation, Management, and Security, CRC Press, 2016.

2. Michael Miller “Cloud Computing Web based application that change the way you work and collaborate online”. Pearson edition, 2008.

3. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More, Jones & Bartlett Learning, 2012.

**WEB REFERENCES**

https://www.ibm.com/cloud

https://www.javatpoint.com/cloud-computing-tutorial

**COURSE OUTCOMES**

Upon Completion of the syllabus the students are able to know:

* Introduce the broad perceptive of cloud architecture and model.
* Understand the concept of parallel and distributed computing
* Understand the different technologies.
* Understand the features of virtualization.
* Learn to design the trusted cloud Computing system with different cloud platform

## CORE ELECTIVE

## PAPER - 4

## C. BIG DATA ANALYTICS

**COURSE OBJECTIVES**

* To understand the needs for Big Data and its environments.
* To learn the basic requirements of Big Data Technologies.
* To expose the knowledge of MapReduce programming framework(Hadoop).
* To be familiar with with NoSQL DB’s Cassandra and MongoDB
* To understand Hive and Pig technologies for analyzing the Big Data.

**UNIT – I: INTRODUCTION TO BIG DATA**

Data, Characteristics of data and Types of digital data: Unstructured, Semi-structured and Structured, Sources of data, Working with unstructured data, Evolution and Definition of big data, Characteristics and Need of big data, Challenges of big data, Data environment versus big data environment

**UNIT – II: BIG DATA ANALYTICS**

Overview of business intelligence, Data science and Analytics, Meaning and Characteristics of big data analytics, Need of big data analytics, Classification of analytics, Challenges to big data analytics, Importance of big data analytics, Basic terminologies in big data environment

**UNIT – III: BIG DATA TECHNOLOGIES AND DATABASES**

Introduction to NoSQL, Uses, Features and Types, Need, Advantages, Disadvantages and Application of NoSQL, Overview of NewSQL, Comparing SQL, NoSQL and NewSQL, Introduction to MongoDB and its needs, Characteristics of MongoDB, Introduction of apache cassandra and its needs, Characteristics of Cassandra

**UNIT – IV: HADOOP FOUNDATION FOR ANALYTICS**

History, Needs, Features, Key advantage and Versions of Hadoop, Essential of Hadoop ecosystems, RDBMS versus Hadoop, Key aspects and Components of Hadoop, Hadoop architectures

**UNIT – V: HADOOPMAPREDUCE AND YARN FRAMEWORK:**

Introduction to MapReduce, Processing data with Hadoop using MapReduce, Introduction to YARN, Components, Need and Challenges of YARN, Dissecting YARN, MapReduce application, Data serialization and Working with common serialization formats, Big data serialization formats

**TEXT BOOK**

Seema Acharya and Subhashini Chellappan, “Big Data and Analytics”, Wiley India Pvt. Ltd., 2016

**REFERENCE BOOKS**

1. “Big Data” by Judith Hurwitz, Alan Nugent, Dr. Fern Halper and Marcia Kaufman, Wiley Publications, 2014.

2.“Big Data Imperatives : Enterprise Big Data Warehouse, BI Implementations and Analytics” by Soumendra Mohanty, Madhu Jagadeesh and Harsha Srivatsa, Apress Media, Springer Science + Business Media New York, 2013

3. “Mining of Massive Datasets”, Anand Rajaraman, Jure Leskovec, Jeffery D. Ullman, Springer, July 2013.

4. “Hadoop: The definitive Guide”, Tom White, O'Reilly Media, 2010.

**WEB REFERENCES**

http://strata.oreilly.com/2010/09/the-smaq-stack-for-big-data.html http://blogs.computerworld.com/18840/big\_data\_smaq\_down\_storage\_mapreduce\_and\_query

**COURSE OUTCOMES**Upon completion of the course, the students will be able to:

* Learn about types of digital data and big data
* Gain knowledge of various Big data analtics and its Technologies
* Study about various NoSQL databases and management techniques
* Work with NoSQL databases such as MongoDB and Cassendra
* Design Big data queries using Hive and Pig.

## OPEN ELECTIVE

## PAPER - 4

## (to choose one out of 3)

## A. INTRODUCTION TO DATABASE SYSTEM

## COURSE OBJECTIVES

* To have a broad understanding of database concepts and database management system software
* To have a high-level understanding of major DBMS components and their function
* To be able to model an application’s data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
* To be able to write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.
* To be able to program a data-intensive application using DBMS APIs.

**UNIT-I: INTRODUCTION**

File System Vs. DBMS - Database System Applications - View of Data-Database language - Database design - ER Model \_ Relational Model - Network Data Model - Hierarchical Data Model - Data Storage & Querying - Data Architecture.

**UNIT-II: RELATIONAL MODEL**

Relational Model - Structure of Relational Databases - Relational Algebra and Calculus - SQL - Basic Structure - Set Operations - Aggregate Functions - Null Values - Nested Queries - Complex Queries - Views - Modification of the Database - Advanced SQL - Triggers.

**UNIT-III: FUNCTIONAL DEPENDENCIES**

Functional Dependencies - Features of Relational designs - Decomposition and Normalization using Functional Dependencies and Multivalued Dependencies - Join dependencies- Domain key Normal form.

**UNIT- IV: PHYSICAL STORAGE MEDIA**

Overview of Physical Storage Media - Magnetic disks - RAID - tertiary Storage - File Organization - Organization of records in Files - Indexing and Hashing - Ordered Indices - B+ -Tree Index Files - B-Tree Index Files - multiple Key Access - Static and Dynamic Hashing - Query Processing - Transaction Management - Transactions - Concurrency.

**UNIT-V: DISTRIBUTED DATABASES**

Distributed Databases - Homogeneous and Heterogeneous Databases - Distributed Data Storage - Distributed Transactions - Commit Protocols - Concurrency Control - Object Based Databases - Complex Data types - Structured Types and Inheritance in SQL – Object identity and Reference - Types in SQL - XML - structure of XML data - XML Document - Schema - Querying and Transformation - Data Mining and Data Warehousing.

**TEXT BOOK**

Abraham Silberschatz, Henry F. Korth and S. Sudarshan- “Database System Concepts”, FifthEdition,McGraw-Hill,2006.

**REFERENCES**

1. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, Tata McGraw-Hill Publishing Company, 2003.

2. RamezElmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003.

3. Hector Garcia–Molina, Jeffrey D.Ullman and Jennifer Widom- “Database System Implementation”- Pearson Education- 2000.

4. Narang,”Database Management Systems”, 2nd ed., PHI.

**WEB REFERENCES**

https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

http://www.rjspm.com/PDF/BCA-428%20Oracle.pdf

http://kadakiaeducation.edu.in/Course/BCA/Course%20Material/RDBMS.pdf

**COURSE OUTCOMES**

On completion of this course students are able to:

* Have a broad understanding of database concepts and database management system software
* Have a high-level understanding of major DBMS components and their function
* Model an application’s data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
* Write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.
* Program a data-intensive application using DBMS APIs.

## OPEN ELECTIVE

## PAPER - 4

## B. INTRODUCTION TO IOT

**OBJECTIVES**

## To design and Develop IOT based solution for real world applications

## To realize the evolution of Internet in Mobile Devices, Cloud & Sensor Networks

## To understand the building blocks of Internet of Things and its characteristics.

## To understand the concepts of IOT and its application.

**UNIT - I: INTRODUCTION**

Introduction and Definition of Internet of Things, IoT Growth – A Statistical View, Application Areas of IoT, Characteristics of IoT, Things in IoT, IoT Stack, Enabling Technologies, IoT Challenges, IoT Levels, Is Cyber Physical System the same as IoT? Is WSN the same as IoT?

**UNIT - II: INTRODUCTION TO SENSORS, MICROCONTROLLERS, AND THEIR INTERFACING**

Introduction to Sensor Interfacing, Types of Sensors, Controlling Sensors through Webpages, Microcontrollers: A Quick Walkthrough, ARM. Protocols for IoT – Messaging and Transport Protocols, Messaging Protocols (MQTT, CoAP, AMQP), Transport Protocols (Li-Fi, BLE).

**UNIT - III: PROTOCOLS FOR IOT**

Addressing and Identification, Internet Protocol Version 4 (IPv4), Internet Protocol Version 6 (IPv6), Uniform Resource Identifier (URI). Cloud for IoT - Introduction, IoT with Cloud – Challenges, Selection of Cloud Service Provider for IoT Applications: An Overview, Introduction to Fog Computing, Cloud Computing: Security Aspects, Case Study: How to use Adafruit Cloud? Application of Data Analytics in IOT.

**UNIT - IV: APPLICATION BUILDING WITH IOT**

Introduction, Smart Perishable Tracking with IoT and Sensors, Smart Healthcare – Elderly Fall Detection with IoT and Sensors, Smart Inflight Lavatory Maintenance with IoT, IoT–Based Application to Monitor Water Quality, Smart Warehouse Monitoring – Let the Drone Fly for You, Smart Retail – IoT Possibilities in the Retail Sector, Prevention of Drowsiness of Drivers by IoT-Based Smart Driver Assistance Systems, System to Measure Collision Impact in an Accident with IoT.

**UNIT - V: GETTING FAMILIARIZED WITH ARDUINO IDE**

Architecture, Arduino Programming, A Simple Application, Arduino Playground. Getting Familiarized with Raspberry Pi - Story behind Raspberry Pi, Architecture, Compatible Peripherals, Add-Ons, and Accessories, Operating System for Raspberry Pi, Setting up Raspberry Pi, Initial Configuration for Raspberry Pi, Linux Based Softwares in Raspberry Pi, Application Development with Raspberry-Pi – A Quick Walk Through.

**TEXT BOOK**

1. Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, Internet of Things, Wiley, India, 2019.

**REFERENCES**

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on Approach)”, 1stEdition, VPT, 2014.
2. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013.

**WEB REFERENCES**

1. https://www.coursera.org/courses?query=iot
2. https://online.stanford.edu/courses/xee100-introduction-internet-things

## https://www.tutorialspoint.com/internet\_of\_things/index.htm

## OUTCOMES

## By the end of the course, the student shall be able to

## Design and Develop IOT based solution for real world applications

## Realize the evolution of Internet in Mobile Devices, Cloud & Sensor Networks

## Understand building blocks of Internet of Things and its characteristics.

## Understand the concept of IOT and its application.

## OPEN ELECTIVE

## PAPER - 4

## C. INTRODUCTION TO MOBILE APPLICATION

**COURSE OBJECTIVES**

* To know the basis of Android application and development environment
* To able to develop simple and professional application
* To get ready for the job opportunity in mobile application development

**UNIT - I: INTRODUCTION TO ANDROID**

History of Android Platform- Android APIs- Android Architecture Application Framework- Features of Android- Android Applications- Application Components - Manifest File- Downloading and Installing Android and Android SDK - Setting up Android Virtual and physical Device - Exploring the Development Environment - The Java Perspective Using Eclipse - DDMS Perspective - Command-Line Tools- Developing and Executing the First Android Application - Using Eclipse IDE to Create an Application - Running Your Application - Exploring the Application - Using Command - Line Tools.

**UNIT** – **II: ACTIVITIES, INTENTS AND FRAGMENTS**

Working with Activities- Creating an Activity- Starting an Activity – Managing the Life cycle of an Activity - Applying Themes and Styles to an Activity- Displaying a Dialog in the Activity - Hiding the title of the activity- Using Intents-Exploring Intent Objects- Exploring Intent Resolution- Exploring Intent Filters - Resolving Intent Filter Collision - Linking the Activities Using Intent - Obtaining Results from Intent – Passing Data Using an Intent Object- Fragments - Hiding Title Bar and Screen Orientation - Fragment Implementation - Finding Fragments - Adding, Removing and Replacing Fragments - Finding Activity Using Fragment - Using the Intent Object to Invoke Built-in Application..

**UNIT - III: UI USING VIEWS AND VIEW - GROUPS**

Working with View Groups – Linear Layout – Relative Layout – Scroll Layout – Table Layout – Frame Layout – Tab Layout using the Action Bar – Working with Views – Text – Edit Text – Button – Radio Button – Check Box – Image Button – Toggle Button – Rating Bar – Binding Data with Adapter View Class – List View – Spinner – Gallery – Designing the Auto Text Complete View – Screen Orientation – Anchoring the Views of Current Activity – Handling UI Events – Handling User Interaction with Activities and Views – Specialized Fragments – List Fragment – Dialog Fragment – Preference Fragment – Creating Menus, Option Menus, Context Menu and Sub Menu.

**UNIT - IV: HANDLING PICTURES AND MENUS WITH VIEWS AND STROING THE DATA**

Working with Image Views – Displaying Images in the Gallery View – Displaying Images in the Grid View – Using the Image Switcher View- Designing Context Menu for Image View- Using the Analog-Clock and Digital Clock Views – Embedding Web Browser in an Activity - Notifying the User Creating the Toast Notification - Creating the Status Bar Notification- Creating the Dialog Notification - Introducing the Data Storage Options - Using Preferences - Using the SQLite Database Creating the Database - Executing the Database Operations.

**UNIT - V: EMAILING, TELEPHONY AND SMS IN ANDROID**

Building an Application to Send Email - Handling Telephony - Displaying Phone InformationApplication Receiving Phone Calls – Making Outgoing Phone Calls Application - Handling SMS Sending SMS Using SMS Manager - Sending SMS Using Intent - Receiving SMS Using the Broadcast Receiver Object- Role of Default SMS Providers - . Publishing Android Application: Export android application – Google play store registration.

Supplementary Learming: Building Mobile Applications using Xamarin

**TEXT BOOKS**

1. Pradeep Kothari, “Android Application Development (with kitkat support) Black Book”, Kogent Learning Solution Inc., Dreamtech Press India Pvt. Ltd, Wiley Publications.
2. Sayed Y. Hashimi, SatyaKomatineni, Dave MacLean, “Pro Android 2”, 2010 Edition, Wiley publications.

**REFERENCES**

1. Reto Meier ,”Professional Android Application Development”,2009 Edition, Willy

Publication.

1. ZigurdMednieks, Laird Dornin, G. Blake Meike,and Masumi Nakamura, “Programming Android”, OReilly publications.

**WEB REFERENCES**

www.tutorialspoint.com

www.javatpoint.net

www.mkyong.com

[www.java2s.com](http://www.java2s.com)

**COURSE OUTCOMES**

After the completion of the syllabus the student will be able to:

* Know about the mobile application development environment
* Develop interface and design
* Use the techniques in Mobile Applications

\*\*\*\*\*\*\*\*\*\*\*\*\*