

**Parvatibai Chowgule College of Arts and Science**  
**(Autonomous)**  
**Margao, Goa**

B.Voc.(Software Development)  
(2020-2021)

Offered to students taking admission to First Year B.Voc from **2019-20**

Semester	General Education Component			Skill Component		
		T	P		T	P
I	Language Paper I: CSD-GE1	4	0	Computer Organization and Operating System CSD-SK1	3	3
	Elements of Basic Statistics CSD-GE2	2	0	Web Design CSD-SK2	3	3
	Cyber Security CSD-GE3	4	0	Introduction to Programming CSD-SK3	3	3
II	Office Automation Tools CSD-GE4	4	0	Database Management Systems CSD-SK4	3	3
	Mathematical foundation of Computer Science CSD-GE5	4	0	Content Management System CSD-SK5	3	3
	Academic Writing CSD-GE6	4	0	Multimedia CSD-SK6	3	3
	Elements of Basic Statistics II CSD-GE7	2	0			
III	Environmental Studies-I CSD-GE8	2	0	Object Oriented Programming CSD-SK7	3	3
	Business Communication CSD-GE9	4	0	Computer Networks CSD-SK8	3	3
	Accounting for Non-accountants CSD-GE10	4	0	Server Side Programming CSD-SK9	3	3
	Internship	2				
IV	Entrepreneurship CSD-GE11	4	0	Web Development Framework CSD-SK10	3	3
	Environmental Studies-II CSD-GE12	2	0	Software Engineering CSD-SK11	3	3
	Personality Enhancement CSD-GE13	4	0	Mobile Application Development CSD-SK12	3	3
	Internship	2				
V	Digital Marketing CSD-GE14	4	0	Data Structures CSD-SK13	3	3
	Organization Behavior CSD-GE15	4	0	Software Testing CSD-SK14	3	3
	Math for Competitive Exams CSD-GE16	4		Project Work	6	
VI	E-commerce CSD-GE17	4	0	Network Security CSD-SK15	3	3
	Independent Studies CSD-GE18	4	0	Cloud Computing CSD-SK16	3	3
	Human Compute Interactions CSD-GE19	4	0	Project	6	

Offered to students taking admission to First Year B.Voc from **2018-19**

Semester	General Education Component			Skill Component		
		T	P		T	P
I	Language Paper I: COM- I.SD-G1	4	0	Office Automation Tools COM- I.SD-SK1	3	3
	Elements of Basic Statistics COM- I.SD-G2	2	0	Web Design COM- I.SD-SK2	3	3
	Cyber Security COM- I.SD-G3	4	0	Introduction to Programming COM- I.SD-SK3	3	3
II	Language Paper II: COM- II.SD-G4	4	0	Data Structure COM-II.SD-SK4	3	3
	Mathematical foundation of Computer Science COM-II.SD-G5	4	0	Computer Organization and Operating System COM-II.SD-SK5	3	3
	Academic Writing COM-II.SD-G6	4	0	Multimedia COM-II.SD-SK6	3	3
	Elements of Basic Statistics II COM-II.SD-G7	2	0			
III	Environmental Studies-I COM-III.SD-G8	2	0	Object Oriented Programming COM-III.SD-SK7	3	3
	Business Communication COM-III.SD-G9	4	0	Computer Networks COM-III.SD-SK8	3	3
	Accounting for Non-accountants COM-III.SD-G10	4	0	Database Management Systems COM-III.SD-SK9	3	3
	Internship	2				
IV	Entrepreneurship COM-IV.SD-G11	4	0	Web Development Framework COM-IV.SD-SK10	3	3
	Environmental Studies-II COM-IV.SD-G12	2	0	Agile Software Engineering COM-IV.SD-SK11	3	3
	Personality Enhancement COM-IV.SD-G13	4	0	Mobile Application Development COM-IV.SD-SK12	3	3
	Internship	2				
V	Digital Marketing COM-V.SD-G14	4	0	Design Analysis of Algorithms COM-V.SD-SK13	3	3
	Organization Behavior COM-V.SD-G15	4	0	Software Testing COM-V.SD-SK14	3	3
	Math for Competitive Exams COM-V.SD-G16	4		Project Work	6	
VI	E-commerce COM-VI.SD-G17	4	0	Content Management System COM-VI.SD-SK15	3	3
	Independent Studies COM-VI.SD-G18	4	0	Cloud Computing COM-VI.SD-SK16	3	3
	Human Compute Interactions COM-VI.SD-G19	4	0	Project	6	

**Course Title:** Computer Organization And Operating System

**Course Code:** CSD-SK1

**Marks:** 75

**Credits:** 3

**Duration:** 45 hrs

: Nil

**Course Objectives:**

1. To have a thorough understanding of the basic structure and operation of a digital computer.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Describe Von Neumann architecture.

CO2: Explain Basic Structure, Function and Operation of a digital computer.

CO3: Explain Memory Subsystem in a computer.

CO4: Describe the function of a Processor, Memory, I/O and System Bus.

CO5: State the difference between various types of Operating Systems.

CO6: Explain the role an OS plays in Memory, Processor and Storage Management.

## **UNIT 1**

[15hrs]

### **Computer System**

Function and structure of a computer, Interconnection of components, Performance of a computer.  
Computer Architecture – Princeton (Von Neumann) and Harvard architecture

### **Memory Subsystem:**

Characteristics of memory system, the memory hierarchy, Semiconductor memories, Types of ROM & RAM, Cache memory unit - Concept of cache memory, Organization of a cache memory unit, replacement algorithms, write policy, block size.

### **Input/ Output Subsystem:**

General block diagram of External device & I/O module, Programmed I/O, Interrupt driven I/O, DMA, I/O channels and I/O processors. I/O interfaces –Serial port, Parallel port, PCI bus, SCSI bus, USB bus, Firewire and Infiniband.

## **UNIT 2**

[15hrs]

### **Introduction to Operating System:**

Basic elements of a computer system: Processor, Main Memory, I/O Modules, System Bus, Instruction Execution; Operating Systems: Definition, Operating system Structure, operating system operations, Relationship between Kernel, OS, and Hardware, Operating system services, System calls, Types of system calls, System programs.

### **Process Management:**

Process Definition, Process Control Block, Process States, Operations on Process; Interprocess communication, Threads and Microkernels.

### **UNIT 3**

[15hrs]

### **Memory Management:**

Introduction, Swapping, Contiguous Memory Allocation, Paging, Page Table, Segmentation, Virtual Memory: Introduction, Demand Paging, Page Replacement, Allocation of Frames, Thrashing

### **Storage Management**

File System, Concepts, File Organization and Access Methods, Directory and Disk Structure. Secondary Storage Structure - Overview, disk structure, Disk attachment, Disk scheduling

### **REFERENCES**

#### **Mandatory Reading:**

1. William Stallings (2013). *Computer Organization and Architecture* (9th ed) Pearson
2. William Stallings (2019). *Operating Systems: Internals and Design Principles* (9th ed) Pearson

#### **Supplementary Reading:**

1. Andrew S. Tanenbaum (2016). *Operating Systems: Internals and Design Principles* (4th ed) Pearson Education India

#### **Web References:**

1. [https://www.tutorialspoint.com/operating\\_system/index.htm](https://www.tutorialspoint.com/operating_system/index.htm)
2. <https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/>
3. [https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system)

**Practical:** Computer Organization And Operating System

**Marks :** 75

**Duration:** 45 hrs

**Credits:** 3

### **PART-I**

[5P]

1. Exploring the Functions and Components of a PC
  - a) Identifying each components
  - b) Learning their function.
2. Building a PC from Scratch.
  - a) Installing the motherboard
  - b) Identifying the CPU socket type and motherboard compatibility.
  - c) Installation of and RAM and its comparability with the motherboard.
  - d) Power Supply requirements and installation.
  - e) Cable management.
3. BIOS ROM Setup.
  - a) Changing the boot priority
  - b) Understanding UEFI and Legacy boot.
  - c) Setting a bios password

**PART-II**

[10P]

1. Partitioning and formatting disk
2. Making a flash drive/ CD ROM bootable.
3. Installing an OS
4. Installing applications device
5. Configuring Display Settings I.e multiple displays
6. Printer setup and configuration
7. Creating, modifying and deleting user account
8. Dual booting with windows
9. Adding and removing software
10. Manually Mounting Partitions
11. Setting up a Virtual OS using Virtual Box
12. Study of Basic commands of Linux.
13. Shell Programming in Unix/Linux, arithmetic operations, loops
14. Menu Driven Shell scripting
15. Filters and Pipes in LINUX

\*\*\*\*\*

**Course Title:** Web Design

**Course Code:** CSD-SK2

**Marks:** 75

**Credits :** 03

**Duration:** 45 hrs

: Nil

**Course objectives:**

1. How to design websites that are responsive.
2. Create interactive web applications using javascript.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Demonstrate the use of various HTML and CSS elements

CO2: Design responsive websites

CO3: Implement frameworks used in web designing.

CO4: Build interactive applications using Javascript

CO5: Apply markup language for presenting of information in web pages

## **SYLLABUS**

### **UNIT 1: HTML and CSS**

[15 Hrs]

HTML Introduction. The development process, basic HTML, formatting and fonts, commenting code, color, hyperlink, lists, tables, images, simple HTML forms, web site structure, Meta tags, Input types, form elements, form attributes, CSS Introduction – Syntax, Id & Class, Backgrounds, Text, Fonts, Box Model – Border, Outline, Margin, Padding. Advanced - Grouping/Nesting, Dimension, Display, Positioning, Floating, Align, Pseudo-class, Pseudo-element, Navigation Bar, Image Gallery, Image Opacity, Media Types, Transitions and Animations.

### **UNIT 2: Javascript**

[15 Hrs]

Introduction - What is JavaScript, Understanding Events, JavaScript Example, External JavaScript. Basic Elements – Comment, Variable, Global Variable, Data Types, Operators, If Statement, Switch, Loop: for and while, Function. JavaScript Objects – objects, Array. Browser Object Model - Browser Objects, Window Object, Document Object – getElementById, getElementsByName, getElementsByTagName, innerHTML property, inner Text property. Validation- form validation, email validation.

### UNIT 3: Bootstrap and jQuery Framework

[15 Hrs]

jQuery Introduction - Syntax, Selectors, Events. Effects- Hide/Show, Fade, Slide, Animate, stop(), Callback, Chaining. HTML/CSS- Add, Remove, CSS Classes, css(), Dimensions, slider.aversing – ancestors, descendants, siblings, filtering, bootstrap components - alert, button, card, carousel. Forms, list groups, modal, navbar, pagination, progress

### REFERENCE

#### Mandatory Reading:

1. Elisabeth R, Eric F, (2012). *Head First HTML and CSS* (2nd ed). Canada, O'Reilly Media, Inc.

#### Supplementary Reading:

1. Ivan B (2017), *HTML5 and CSS3 Made Simple* (1st ed). India, BPB publication
2. DT Editorial Services (2016), —*HTML5 Black Book: Covers CSS3, Javascript, XML, XHTML, Ajax, PHP and JQuery* (2nd ed). India, Dreamtech Press.

#### Web References:

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css/>

<https://www.w3schools.com/jquery/>

<https://www.w3schools.com/bootstrap/>

#### Practical: Web Design

**Marks:** 75

**Duration:** 45 hrs

**Credits:** 03

#### List of Assignments:

Creating a webpage using the following html tags (1P)

Creating a webpage styled using CSS (1P)



Embed Video and audio in an HTML page (1P)

Create a simple navigation bar (1P)

Replicate the footer of a website (1P)

Replicate a website (2P)

Using javascript validate (3P)

Check whether a given email address is valid or not.

Check whether a value is a number or not.

Check whether a given credit card number is valid or not.

Check whether a given credit card number is valid or not.

Using jQuery (3P)

Show and hide html elements

Using Bootstrap (2P)

Create a responsive website using bootstrap

\*\*\*\*\*

**Course Title:** Introduction to Programming

**Course Code:** CSD-SK3

: Nil

**Marks:** 75

**Credits:** 03

**Duration:** 45 hrs

**Course Objective:-**

1. Imbibe basic programming skills and industry level coding standards.

**Course Outcomes:** On completion of the course student will be able to

CO1: Explain the different programming paradigms.

CO2: Write computer program to solve basic computational problems.

CO3: Write computer program that perform basic I/O operations.

CO4: Use python function, modules and exceptions effectively for solving complex problems.

**UNIT 1**

**[15hrs]**

## **Introduction to Python:**

Introduction to programming: History of Programming, Algorithm, Other Programming Languages, Programming Paradigms. Setting up Python: Installation, Python Interpreter, Python IDLE, Running Python Programs.

## **Basic Syntax**

Variable and Data Types, Operators, Conditional Statements: if, if – else, Nested if-else. Looping: For, While, Nested Loops, Control Statements: break, continue, pass.

## **UNIT 2**

**[15hrs]**

### **String:**

Accessing Strings, Basic Operations, String slices, Function and Methods.

### **Lists:**

Introduction, Accessing list, Operations, Working with lists, Function and Methods

### **Dictionaries**

Introduction, Accessing values in dictionaries, Working with dictionaries, Properties, Functions.

## **UNIT 3**

**[15 Hrs]**

### **Functions**

Defining a function, Calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables.

### **Modules**

Importing module. Math module. Random module. Packages.

### **Exception Handling**

Exception. Exception Handling - Except clause, Try, finally clause. User Defined Exceptions

## **REFERENCES**

### **Mandatory Reading:**

1. Martin C. Brown (2018). *Python: The Complete Reference* (4th ed) McGraw Hill Education

### **Supplementary Reading:**

1. Mark Lutz (2013). *Learning Python* (5th ed) O'Reilly

### **Web References:**

1. <https://www.learnpython.org/>
2. <https://www.tutorialspoint.com/python/index.htm>
3. [https://www.youtube.com/watch?v=\\_uQrJ0TkZlc&t=823s](https://www.youtube.com/watch?v=_uQrJ0TkZlc&t=823s)

**Practical:** Introduction to Programming

**Marks :** 75

**Duration:** 45 hrs

**Credits:** 3

1. Installing python and getting familiar with the python IDLE.
2. Basic math operators.
3. Relational and Logical Operators
4. Scope and Indentation
5. If statement
6. While loop
7. For loop
8. Input from keyboard and Type casting
9. File Operations
10. String manipulation
11. Lists
12. Dictionary
13. Modules
14. Functions
15. Exception Handling

\*\*\*\*\*

**Course Title:** Database Management System

**Course Code:** CSD-SK4

**Marks:** 75

**Credits:** 03

**Duration:** 45 hrs

: Nil

**Course objectives:**

1. To develop database model and apply to medium scale application.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Model an application's data requirements using conceptual modeling tools like ER diagrams.

CO2: Apply Normalization theory to normalization a database.

CO3: Use a database management system to create, populate, maintain, and query a Database.

CO4: Analyze a given problem and select an appropriate database.

**SYLLABUS****UNIT 1: Introduction to database and ER models [15 Hrs]**

File Systems versus DBMS, The Relational Model, Levels of abstraction in a DBMS, Data independence, Structure of DBMS, Advantage of DBMS, People who deal with Databases. Overview of Database Design –The ER model-features, Key Constraints, Participation Constraints, weak Entities, Class Hierarchies, Aggregation.

**UNIT 2: SQL [15 Hrs]**

Attributes and domains, Relations, Integrity Constraints, Key Constraints, Foreign Key Constraints, General Constraints, Query Languages SQL: The Form of Basic SQL query, Condition specification, SQL Join, Union, Intersect, Except, Nested queries - Aggregate Operators, updates, Null values, Embedded SQL, Triggers, Data Definition Language, Introduction to Database Security : views

**UNIT 3: Schema Refinement, Normal forms, transactions and Latest trends [15 Hrs]**

Introduction, Schema Refinement, Functional Dependencies, Closure of a set of FDs and Attribute closure, Normal Form 1NF, 2NF, Third Normal Form, BCNF, Decomposition Lossless-Join Decomposition, Dependency-Preserving Decomposition, Normalisation Decomposition into BCNF, Decomposition into 3NF, The concept of transaction, transaction and schedule, Notion of consistency, NOSQL databases, Spatial Databases, Multimedia Databases, Distributed databases.

## REFERENCE

### Mandatory Reading:

1. Silberschatz A, Korth H F, Sudarshan S (), *Database system concepts* (8th ed). India, McGraw-Hill.

### Supplementary Reading:

1. Ramakrishnan, Gehrke J (2013). *Database management systems* (3rd ed.), McGraw-Hill
2. Elmasri R, Navathe S (). *Fundamentals of database Systems* (6th ed.). India, Pearson Education

### Web References:

<https://www.tutorialspoint.com/sql/index.htm>

<https://www.w3schools.com/sql/>

<https://dev.mysql.com/doc/>

<https://www.guru99.com/sql.html>

### Practical: Database Management Systems

**Credit:** 3

**Duration:** 45 hrs

**Marks:** 75

#### List of Practicals

1. ER diagram (2P)
2. Create database and tables (2P)
3. Basic SQL queries (2P)
4. Joins (2P)
5. Transactions (2P)
6. Python Database API (2P)
7. Mini project (3P)

\*\*\*\*\*

**Course Title:** Multimedia

**Course Code:** CSD-SK6

**Marks:** 75

**Credits:** 03

**Duration:** 45 hrs

Nil

**Course Objectives:**

1. Develop specific skills and competencies by making them proficient in Designing Graphical Images, Audio and Video Capture and Editing using Software tools

**Course Outcomes:**

CO1: Develop specific skills in designing Graphical Images, Audio and Video Capture and Editing using Software tools

CO2: Explain the industrial standard of video, audio and image formats.

CO3: Explain where and when to use image manipulation software tools.

CO4: Describe the process of editing audio/video/image content.

UNIT 1:

[15hrs]

**Introduction to Multimedia :**

Commonly used terms associated with multimedia like CDROM, Storyboard, Script and Authoring tools. Stages of a Multimedia Project: Planning and Costing, Designing and Producing, Testing and Delivering. The Multimedia team and their roles: Project Manager, Writer, Video specialist, Audio specialist and Multimedia programmer. Multimedia Software. Multimedia Hardware.

**Multimedia Authoring Tools:**

Types of Authoring tools - card or page based tools, icon-based, event-driven tools, time-based and presentation tools and object-oriented tools.

UNIT 2:

[15hrs]

**Multimedia Building Blocks:**

Designing with Text, menus and buttons for navigation, Animating text, Hypermedia and Hypertext Basic Sound Concepts, Music, Speech, MIDI and Digital Audio Making still images, Bitmaps, Clipart, Capturing and Editing Images, Scanning Images, Vector, Drawing, 3D Drawing and Rendering.

**Animation**

Principles of Animation- persistence of vision, animation file formats Computer animation, kinematics and morphing, Making animations that work- a rolling ball, a bouncing ball and creating an animated scene.

UNIT 3:

[15hrs]

## **Video**

Video Broadcast Standards- NTSC, PAL, SECAM, HDTV , Integrating Computers and, Television like Video Overlay Systems, Digitized Video Playback, , Differences between Computer and Television Video , Recording Formats like S-VHA Video, Component (YUV), Component Digital, Composite , Digital, Video Hardware Resolutions , Video Tips like Shooting platforms, Lighting, Chroma Key or Blue Screen , Video Compression methods like MPEG and DVI.

## **Assembling and Delivering a project**

The four primary navigational structures used in multimedia like linear, hierarchical, non-linear and composite.

## **REFERENCES**

### **Mandatory Reading:**

1. Vaughan, T. (2011). *Multimedia: Making it work*. McGraw-Hill.

### **Supplementary Reading:**

1. Vic, C. (2016). *Multimedia Foundations: Core Concepts for Digital Design*. Routledge

### **Web References:**

1. <https://www.youtube.com/watch?v=aCisC3sHneM>
2. <https://www.gimp.org/tutorials/>
3. <https://www.infotrendnow.com/2018/08/openshot-tutorial.html>

**Practical:** Multimedia

**Credit:** 03

**Duration:** 45 hrs

**Marks:** 75

## **List of PRACTICALS**

1. Image Handling: [1P]

Cropping an image, adjusting image size, increasing the size of the work canvas, saving an image

2. Layers: [2P]

Adding layers, dragging and pasting selections on to layers, dragging layers between files, viewing and hiding layers, Editing layers, rotating selections, scaling an object, preserving layers transparency, moving and copying layers, duplicating layers, deleting layers, merging layers, using adjustment layers.

3. Channels and Masks: [2P]

Channel palette, showing and hiding channels, splitting channels in to separate image, merging channels, creating a quick mask, editing masks using quick mask mode

4. Painting and Editing: [2P]

Brushes palette, brush shape, creating and deleting brushes, creating custom brushes, setting brush options, saving, loading and appending brushes, Options palette.

5. Image Editing Effects and Tools: [2P]

Opacity, pressure, or exposure , paint fade-out rate, making selections, using selection tools, adjusting selections, softening the edges of a selection, hiding a selection border, moving and copying selections, extending and reducing selections, pasting and deleting selections

6. Sound : [2P]

Recording Sound using Sound Recorder (Capture), Sound capture through sound editing software , Sound editing, Noise correction, Effect enhancement ; Voice Recognition; Importing audio and saving audio from Audio CD. Sound Quality types: CD Quality, Radio Quality, Telephone Quality

7. Video: [2P]

Record video from video capture devices, webcams, screen capture or even streaming video and Video Editing

8. Mini Project/Problem Statement/Case Study (integrating the above experiments) [2P]

\*\*\*\*\*

**Course Title :** Object Oriented Paradigm

**Course Code :** CSD-SK7

**Marks :** 75

**Credits :** 3

**Duration:** 45 hrs

**:** Nil

**Course Objectives:**



- To learn the basic concepts and techniques of object oriented programming paradigm
- To introduce object oriented programming (OOP) using Java.

### **Course Outcomes:**

On successful completion of this course students will be able to:

CO1 : Explain the benefits of object oriented design and understand when it is an appropriate methodology to use.

CO2 :Differentiate between the top-down and bottom-up approach.

CO3:Develop problem-solving and programming skills using the OOP concept.

CO4 :Apply the concepts of object-oriented programming using Java.

CO5: Design object oriented solutions for small systems involving multiple objects.

CO6: Develop GUI for an application.

### **SYLLABUS:**

#### **Unit 1:**

**[15Hrs]**

#### **Principles of OOP**

Programming Paradigms, Basic concepts, OOP: major principles - encapsulation, abstraction, inheritance, polymorphism. Benefits of OOP, Applications of OOP.

#### **Introduction to Java**

Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, java.Math class, Arrays in java.

#### **Unit 2:**

**[15Hrs]**

#### **Objects and Classes**

Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, StringBuffer, File, this reference.

#### **Inheritance and Polymorphism.**

Inheritance in java, super and sub class, Overriding, java.lang.Object class, Polymorphism, Dynamic binding, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, java.util package.

### **Unit 3:**

**[15Hrs]**

#### **Event driven and GUI programming**

Windows and Layout Manipulation, Dialogs (Message, confirmation, input), Event Handling: Event sources, Listeners, Mouse and Keyboard Event Handling.

#### **Exception Handling**

Exception handling – what and why? Try and catch block. Multiple catch blocks. Nested try, finally block, throw keyword, throws keyword. Custom Exception.

#### **Multithreading**

Running and starting thread using Thread class. Thread priorities. Running multiple threads. The Runnable interface. Synchronization and inter thread communication.

#### **Mandatory Reading:**

1. Matha, M. P. (2011). *Core Java: a comprehensive study*. PHI Learning.

#### **Supplementary Reading:**

1. Balaguruswamy, E. (2014). *Programming with Java-A Primer*. McGraw-Hill Professionals.

#### **Web References:**

1. [www.tutorialspoint.com/java/index.htm](http://www.tutorialspoint.com/java/index.htm)
2. [https://www.w3schools.com/java/java\\_intro.asp](https://www.w3schools.com/java/java_intro.asp)
3. <https://www.geeksforgeeks.org/java/>

**Practical:** Object Oriented Paradigm

**Credit:** 3

**Duration:** 45 hrs

**Marks:** 75

Programs using Java language that covers the following concepts:

- |  |      |
|--|------|
| 1. Classes and instances                         | (2P) |
| 1. Working with the java.Math class              | (1P) |
| 2. Inheritance                                   | (1P) |
| 3. Polymorphism, abstract classes and interfaces | (2P) |
| 4. Utilising the java.util package               | (1P) |
| 5. Collections framework                         | (2P) |
| 6. Event handling and GUI                        | (2P) |
| 7. Exception handling                            | (2P) |
| 8. Mini Project                                  | (2P) |

\*\*\*\*\*

**Course Title:** Computer Networks

**Course Code:** CSD-SK8

**Marks:** 75

**Credits:** 3

**Duration:** 45 hrs

**Prerequisite Courses:** Nil

**Course Objectives:**

1. Gain Knowledge of the Reference models
2. Understand basic concepts of data transmission medium, Compare various routing, transport protocols and Identify suitable protocol for a given network.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Select the most appropriate networking architecture and technologies for the given organization structure.

CO2: Compare and contrast the layers in OSI model and TCP/IP.

CO3: Explain the functionalities provided by each layer in the OSI model.

CO4: Define the concept used for error handling in Datalink layer

CO5: Develop client server programs for different applications.

CO6: Design basic computer network

## **SYLLABUS**

### **UNIT 1: Introduction and Data Link Layer**

[15 Hrs]

Basics of Computer Networks, Classification: transmission technology, scale; Applications; Data Communications: data, signal, bandwidth, bit interval and bit rate, Modes of Communication. Layered network architecture, Networks models: OSI model, TCP / IP protocol suite; Guided and Unguided Transmission media, Multiplexing: FDM, TDM. Switching: Circuit switching, message switching, Packet Switching. Data link control: Framing: Character Count, Character Stuffing, Bit Stuffing , Error Detection and correction, Flow and error control, HDLC; Multiple access: Random access – Controlled access , ALOHA, CSMA, CSMA/CD and CSMA/CA; Ethernet : IEEE standards, standard Ethernet, Fast Ethernet, Gigabit Ethernet; Connecting devices: repeater/hub, bridge, router and gateway, Backbone networks - Virtual LANS.

### **UNIT 2: Network Layer**

[15 Hrs]

Functions of Network layer; Network Service types: Virtual Circuits, Datagrams; Logical addressing: IPv4, private and public IP addressing, special IP addresses, subnetting, IPV6 addressing Internet Protocol: Internetworking:IPv4, Fragmentation and reassembly , Address mapping : ARP, RARP, BOOTP, DHCP, ICMP . Routing: classification of routing, Shortest path routing, Distance Vector routing, Link State routing 4. Transport layer and Application layer

### **UNIT 3: Transport and Application Layer**

[15 Hrs]

Process-to-Process delivery: User Datagram Protocol (UDP), Transmission Control Protocol(TCP), Quality of services (QoS); Application Layer: Domain Name System (DNS) , E-mail,FTP, HTTP.5. Wireless Networks

## **REFERENCE**

### **Mandatory Reading:**

1. Andrew T, David J (2013). *Computer Networks* (5th ed.). India, Prentice-Hall,

### **Supplementary Reading:**

1. Behrouz A (2011), *Data communication and Networking* (4th ed.). Amarica McGraw Hill Education.

2. James F, Keith R (2009). *Computer Networking - A Top-Down Approach Featuring the Internet* (5th ed.). Amarica, Pearson Education

### **Web References:**

<https://www.javatpoint.com/computer-network-tutorial>

<https://www.geeksforgeeks.org/computer-network-tutorials/>

<https://www.studytonight.com/computer-networks/>

<https://www.softwaretestinghelp.com/computer-networking-basics/>

**Practical:** Computer Networks

**Credits:** 3

**Duration:** 45 hrs

**Marks:** 25

List of Practicals

1. Installing virtual machines, Ethernet cabling [1P]
2. Study of network commands ping, ipconfig, netstat, traceroute [1P]
3. Setting up of LAN Network [1P]
4. IP address manipulation -Extract network id and Host id given netmask [1P]
5. Mini Project / Packet capture tool/ packet generator tool [1P]
6. UDP Socket programming (c/c++/Java/ Perl/Python ) [1P]
7. TCP Socket programming [1P]
8. Configuring routing tables [1P]
9. Configuring DHCP server/client [1P]
10. Configuring Telnet/SSH and ftp server. [1P]
11. Firewall Configuring [2P]
12. Write (c/c++/Java/ Perl/Python ) program to implement Bit and byte stuffing. [3P]

\*\*\*\*\*

**Course Title:** Server Side programming

**Course Code:** CSD-SK9

**Marks:** 75

**Credits:** 03

**Duration:** 45hrs

**Course Prerequisite:** Object Oriented Programming

**Course Objectives:**

1. Provide an in depth understanding of a server side language, and use it to develop applications
2. Design and implement basic server-side scripts.
3. Create responsive and interactive web applications using frameworks

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Explain the core features and functionalities of PHP

CO2: Design interactive web application using core PHP

CO3: Develop basic server side script to interact with users and the database

CO4: Build a web application using laravel framework

CO5: Utilize MVC model

**UNIT 1:**

[15 Hrs]

Static vs. Dynamic web pages, Need for Server Side technologies, Multi Tier Web Architecture, file architecture of a web server, Variable Types, Constants, Decision Making-If...Else, ElseIf, Loop Types - For, while, do while, foreach, Array, string, functions, Object Oriented Concepts - Class, Object, member variable, member function, Inheritance, Polymorphism, overloading, Data Abstraction, Encapsulation. Constructor, Destructor.

**UNIT 2**

[15 Hrs]

Use of PHP Tags, Tag Styles, Calling Functions, include, Processing GET and POST request, uploading files to server, cookies, sessions, Difference between MySQLi and PDO, database

connection using PDO, CRUD - Create, Read, Update, Delete, records in database, pagination, login, difference between XML and AJAX, defining AJAX array, parse AJAX using JQuery, AJAX request, AJAX response.

### **UNIT 3**

[15 Hrs]

Introduction to Laravel, Routing in Laravel, MVC in Laravel, Caching in Laravel, Event subscribers in Laravel, Package Development, Templates, Creating an Application, Testing in Laravel, Database Configuration, Helpers in Laravel, Laravel Pagination, Laravel Security, Authentication Facade, Validation in Laravel, Eloquent ORM, Artisan Command Line Interface, Deploy Application using Laravel.

### **REFERENCE**

#### **Mandatory Reading:**

1. Leon A, Zee S(2004), *Core PHP Programming* (3rd ed.).Prentice Hall Professional
2. Stauffer, M. (2019). *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. O'Reilly Media.

#### **Supplementary Reading:**

1. Williams, H. E., & Lane, D. (2004). *Web Database Applications with PHP and MySQL: Building Effective Database-Driven Web Sites*. O'Reilly Media, Inc.
2. Brinzarea, B., & Hendrix, A. (2009). *Ajax and PHP: Building modern Web applications*. Packt Publishing Ltd.

#### **Web References:**

<https://www.w3schools.com/php/>  
<https://www.tutorialspoint.com/php/index.htm>  
<https://laravel.com/docs/6.x>  
<https://www.tutorialspoint.com/laravel/index.htm>

**Practical** :Server Side programming

**Marks:** 75

**Duration:** 45hrs

**Credits:** 03

PHP Classes and instances,PHP Controls Structures	[1P]
PHP Array Programming, Inheritance	[1P]
CRUD using PHP database API's.	[3P]
<ul style="list-style-type: none"> <li>Fetch data from a form, validate and insert in the database.</li> <li>Delete data in the database.</li> <li>Update data in the database</li> <li>Display data from the database.</li> </ul>	
Uploading files and session management.	[1P]
Implementing MVC	[2P]
Migrations in Laravel	[1P]
Using Forms and Gathering Input in Laravel	[1P]
Creating a registration & user login form in Larvael	[1P]
Using Controllers and Routes for URLs and APIs in Laravel	[1P]
Eloquent ORM in Laravel	[1P]
Creating and Using Composer Packages	[1P]
Security & Session [	1P]

\*\*\*\*\*

**Course Title:** Web Development Framework

**Course Code:**CSD-SK10

**Marks:** 75

**Credits:** 03

**Duration:** 45 hrs

**Prerequisite Course:**

Web Design

Object Oriented Paradigm

Database Management System

**Course Objective:-**

1. Use Web Frameworks and Libraries to develop interactive web applications.



**Course Outcomes:** On completion of the course student will be able to

CO1: Use ReactJS to build rich and interactive front end applications.

CO2: Use NodeJS to develop back end application to accept POST,GET,PUT,DELETE requests.

CO3: Develop REST API's using NodeJS.

CO4: Write non-blocking and blocking JavaScript code.

CO5: Explain Framework and Libraries with respect Web Development.

## **UNIT 1**

**[15Hrs]**

### **ReactJS**

History of front end libraries, Motivation for using React, Thinking in React, One way binding, JSX + CSS modules, Virtual DOM, ES6

### **ReactJS:components**

Component lifecycle, Component API, Render functions, State, Props, Mixins

## **UNIT 2**

**[15Hrs]**

### **ReactJS:Interaction between components**

Passing data from parent to child, Passing data from child to parent, Passing data between 2 components at the same level, Forms, Refs, React-Router, API integration

### **NodeJS: Introduction**

Brief overview on the benefits of using Node.js and how Node.js is used in modern web development, Node and NPM, Introduction to setting up a Node.js project, Importing modules using npm, Using core modules to make HTTP requests and manipulate the file system.

## **UNIT 3**

**[15Hrs]**

### **NodeJS: Express framework**

Set up a web server, Implementing API routing, Implementing middle-ware, Implementing URL parameters.

### **NodeJS: MySQL module**

Settings up a database and connecting it to a NodeJS server, Storing and retrieving data from the database.

## **REFERENCES**

**Mandatory Reading:**

1. Brett McLaughlin (2011). *What Is Node ?* (1st ed) O'Reilly Media
2. Alex Banks (2017). *Learning React. (1<sup>st</sup> ed)* Shroff/O'Reilly

**Reference Books:**

1. Mario Casciaro (2016). *Node.js Design Patterns* (2nd ed) Packt Publishing Limited

**Web References:**

1. <https://www.tutorialspoint.com/nodejs/index.htm>
2. <https://reactjs.org/docs/getting-started.html>
3. <https://www.youtube.com/watch?v=Ke90Tje7VS0>

**Practical:** Web Development Framework**Marks:** 75**Duration:** 45hrs**Credits:** 03

1. Creating a simple web server. (1P)
2. Connect to MySQL database. (1P)
3. CRUD using MySQL database API's. (4P)
  - a. Fetch data from a form, validate and insert in the database.
  - b. Delete data in the database.
  - c. Update data in the database
  - d. Display data from the database.
4. Uploading files. (1P)
5. Login functionality using sessions. (1P)
6. Using cookies to store website data. (1P)
7. Mini project. (3P)

\*\*\*\*\*

**Course Title :** Software Engineering**Course Code :** CSD-SK11**Marks :** 75

**Credits : 3**

**Duration:** 45hrs

**Prerequisite Courses:** Nil

**Course Objectives:**

To Understand the various software development methodologies and estimation of software projects

To analyze and design software projects

To study the various phases of a S/W Development Project.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1 :Understand the various Software Development Methodologies

CO2: Apply Estimation techniques to live projects

CO3: Analyze Software Projects.

CO4: Design Software Projects.

**SYLLABUS:**

**Unit 1:**

**[15 Hrs]**

**SOFTWARE PROCESS:**

Characteristics of software process, Software Development Processes and Methodologies: waterfall, prototyping, iterative, spiral, unified process, Benefits of iterative and incremental approach with emphasis on Unified process, CASE Tools, Agile methodologies.

**PROJECT MANAGEMENT:**

Planning a Software Project Cost estimation, Project Scheduling, Software configuration, management plans, Quality Assurance plans, Project Monitoring plans and Risk Management.

Techniques such as Interviewing , Requirement Workshop, brainstorming, prototyping.  
Characteristics of SRS

**Unit 2:**

**[15 Hrs]**

**OOAD and UML:**

OOAD: Definition; object oriented analysis; object oriented design and modeling; Assigning responsibilities.

UML: Main UML diagrams Class diagram , sequence diagram, activity diagram, use case diagram. Use case model use case diagram , use case descriptions, use case realization using sequence and activity diagrams. Supplementary requirements. Advanced use case model features.

Requirements: Functional and non-functional

System Design : Class diagram, sequence diagram, activity diagram, state chart diagram, deployment diagram. Brief introduction to other UML diagrams.

**SOFTWARE ARCHITECTURE PATTERNS:**

Major Architectural Styles (patterns) like Layered Architecture, Pipe and Filter, Shared (Central), Data Store, Event Driven, Model-View-Controller (MVC), Distributed & Emerging Service Oriented Architecture (SOA) and Elementary GRASP Patterns.

**Unit 3:**

**[15 Hrs]**

**HUMAN COMPUTER INTERACTION:**

HCI Definition; User categories, Interface Design-Internal & External Interface design, user interface design, Interface design guidelines.

**CODING:**

Coding styles, standards, peer reviews, checklist.

**TESTING:**

Testing Fundamental, Functional Testing, Structural Testing, Testing Object-Oriented Programs, Testing Process and Metrics.

**DOCUMENTATION and MAINTENANCE:**

Need for Software Documentation. Types of documentation, Need for Maintenance; Types of Maintenance

**RE ENGINEERING:**

Business Process Re engineering, Software Re engineering, Reverse Engineering, Restructuring, Forward Engineering, The Economics of Re engineering

## **References:**

### **Mandatory**

1. Pressman R.S., (2017). *Software Engineering: A Practitioner's Approach*, 6th edition:McGraw Hill
2. Larman C.,(2015). *Applying UML and patterns*, 3rd Edition: Addison Wesley

### **Supplementary**

1. Jalote P., (2010) *An Integrated Approach to Software Engineering*, 3rd Edition :Narosa Publishing House
2. Sommerville I.,(2015) *Software Engineering*, 10th Edition:Adison Wesley
3. Fowler M., (2003) *UML Distilled*, 3rd Edition:Addison Wesley

### **Web References:**

[https://www.tutorialspoint.com/software\\_engineering](https://www.tutorialspoint.com/software_engineering)  
<https://www.w3schools.in/sdlc-tutorial>  
<https://www.geeksforgeeks.org/software-engineering>  
<https://www.javatpoint.com/software-engineering-tutorial>

**Practical** : Software Engineering

**Credit** : 3

**Duration:** 45hrs

**Marks** : 75

List of suggested PRACTICALS :

For a given project/case study

- 1) Requirements Gathering Techniques (2P)
- 2) Gantt Chart (2P)

- 3) USE Case diagram and Use Case descriptions for the Use Cases (3P)
- 4) Class Diagram (2P)
- 5) Sequence Diagram (2P)
- 6) Activity Diagram (2P)
- 7) State Chart Diagram (2P)

\*\*\*\*\*

**Course Title:** Mobile Application Development

**Course Code:** CSD-SK12

**Marks:** 75

**Credits:** 3

**Prerequisite Courses :** Nil

**Course Objective:**

To develop applications for mobile devices, including smart phones and tablets, introduced to the current mobile platforms, mobile application development environments and mobile device input methods.

**Course Outcomes :**

On successful completion of this course students will be able to:

- CO1: Define mobile platforms and their architectures.
- CO2: Compare development for different mobile platforms.
- CO3: Demonstrate the use of Android Components.
- CO4: Develop Mobile applications for Android Platform.
- CO5: Make use of SQLite database.
- CO6: Test Mobile applications for Android Platform.

## **SYLLABUS:**

### **Unit 1:**

**[15Hrs]**

#### **Introduction to mobile devices**

Mobile devices vs. desktop devices, Why we Need Mobile App, Different Kinds of Mobile Apps, ARM and intel architectures, Power Management, Screen resolution, Touch interfaces, Application deployment : App Store, Google Play, Windows Store, Native vs. web applications.

#### **Android Overview**

Introduction to Android. Overview of android stack, Introduction to OS layers, Android features. Linux Kernel, Libraries, Android Runtime, Application Framework, Dalvik VM

#### **Mobile OS Architectures**

Comparing and Contrasting architectures of Android, iOS and Windows, Underlying OS (Darwin vs. Linux vs. Windows ), Kernel structure and native level programming, Runtime (Objective-C vs. Dalvik vs. WinRT), Approaches to power management, Security.

#### **Android Components**

Activities, Services, Broadcast Receivers ,Content Providers.

### **Unit 2:**

**[15Hrs]**

#### **Building UI with Activities**

Activities, Views, layouts and Common UI components, Creating UI through code and XML, Activity life cycle, Intents, Communicating data among Activities.

#### **Advanced UI**

Selection components (GridView, ListView, Spinner ), Adapters, Custom Adapters, Menus, Toast, Custom Toast, Dialogs, Status bar Notifications.

#### **Intent Filters and Broadcast Receivers**

Role of filters, Intent-matching rules, Filters in your manifest, Filters in dynamic Broadcast Receivers, Creating Broadcast receiver, Receiving System Broadcast, Understanding Broadcast action, category and data, Sending Broadcast.

### **Unit 3:**

**[15Hrs]**

#### **Data Storage**

Shared Preferences, Android File System, Internal storage, External storage. SQLite Introducing SQLite, SQLiteOpenHelper and creating a database, Opening and closing a database, Working with cursors, inserts, updates, and deletes

#### **Services**

Overview of services in Android, Implementing a Service, Service lifecycle, Inter Process Communication (AIDL Services). Web Services and WebView - Consuming web services, Receiving HTTP Response (XML, JSON ), Parsing JSON and XML, Using WebView,

#### **Firestore**

Introduction to Firestore and cloud messaging, real time database, authentication.

#### **References**

##### **Mandatory Readings:**

1. Lee, W. M. (2012). *Beginning android 4 application Development*. John Wiley & Sons.

##### **Supplementary Reading:**

1. Burnette, E. (2009). *Hello, Android introducing Google's mobile development platform 2nd*.

##### **Web References:**

<http://developer.android.com>

<https://www.tutorialspoint.com/android/index.htm>

<https://abhiandroid.com/>



**Practical:** Mobile Application Development

**Credit:** 3

**Duration:** 45hrs

**Marks:** 75

**List of practicals**

- |   |      |
|---|------|
| 1. Getting Started with Android                         | (1P) |
| 2. Creating a basic Android Application                 | (2P) |
| 3. Making use of GUI components                         | (2P) |
| 4. Making use of advanced UI components.                | (1P) |
| 5. Implementing Data storage application                | (2P) |
| 6. Implementing Services/Multithreading/Multiprocessing | (2P) |
| 7. Firebase   | (2P) |
| 8. Mini Project   | (3P) |

\*\*\*\*\*

**Course Title:** Data Structure

**Course Code:** CSD-SK13

**Marks:** 75

**Credits:** 03

**Duration:** 45hrs

**Prerequisite:** Introduction to Programming

## **Course Objectives:**

1. To understand different methods of organizing data and efficiently implement different data structures.

## **Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Select appropriate data structures as applied to specified problem definition.

CO2: Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures

CO3: make use of appropriate sorting/searching technique for given problem

CO4: Design advance data structure using NonLinear data structure.

## **SYLLABUS**

### **UNIT 1**

[15 Hrs]

Concept, Data type, Data object, ADT, Need of Data Structure, Types of Data Structure, Stack Introduction, Representation-static & dynamic, Operations, Application - infix to postfix & prefix, postfix evaluation, Simulating recursion using stack, Introduction, Representation -static & dynamic, Operations, Circular queue, priority queue (with implementation), Concept of doubly ended queue.

### **UNIT 2:**

[15 Hrs]

Introduction to List, Implementation of List – static & dynamic representation, Types of Linked List, Operations on List, Applications of Linked List, polynomial manipulation, Generalized linked list – concept & representation, Concept & Terminologies, Binary tree, binary search tree, Representation – static & dynamic, Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes, non-recursive inorder traversal, Expression Tree.

### **UNIT 3:**

[15 Hrs]

Concept & terminologies, Graph Representation – Adjacency matrix, adjacency list, Traversals – BFS & DFS, Application of BFS, DFS – Shortest path, Bubble sort, Merge sort, Selection sort, Sequential Searching, Binary Searching

## **REFERENCE**

### **Mandatory Reading:**

1. Horowitz E, Sahni S (2008), *Fundamentals of Data Structures in C* (2nd ed.). University Press.

**Supplementary Reading:**

1. Langsam Y, Augenstein M, Tenenbaum A (2009). *Data Structures using C and C++* (2nd ed.). Pearson Education

2. Gilbeg R, Forouzan B, *Data Structures: A Pseudocode Approach with C++* (2nd ed.). Cengage Learning

**Web References:**

<https://www.javatpoint.com/data-structure-tutorial>

[https://www.tutorialspoint.com/data\\_structures\\_algorithms/index.htm](https://www.tutorialspoint.com/data_structures_algorithms/index.htm)

<https://www.geeksforgeeks.org/data-structures/>

<https://www.studytonight.com/data-structures/>

**Practical:** Data Structures

**Credit:** 03

**Duration:** 45hrs

**Marks:** 75

-----  
Programs using C language / Java Language that covers the following concepts:

1. Stack: Static/Dynamic stack implementation.
2. Stack: infix to postfix. [2P]
3. Stack: Evaluation of Postfix expression.
4. Queues: Static and Dynamic Queue Implementation
5. Queues: Circular queue
6. List: Singly Linked List, [2P]
7. List: Doubly Linked List [2P]
8. List: Circular Linked List [2P]
9. Linked List: Polynomial addition
10. Trees: Binary Search Tree: create, add, delete, display nodes. [2P]

\*\*\*\*\*

**Course Title:** Software Testing

**Course Code:** CSD-SK14

**Marks:** 75

**Credits:** 3

**Duration:** 45hrs

**Prerequisite Courses:** Nil

**Course Objectives:**

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to plan a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Define Software Testing process for an applications.

CO2: Apply Software Testing process in relation to Software Development and Project Management.

CO3: Create Test Strategies and plans, design test cases, prioritize and execute them.

CO4: Identify various Software Testing problems and solve them.

CO5: Identify the needs of software test automation, and define and develop a test tool to support test automation.

CO6: Use software testing methods and modern software testing tools for their testing projects

**SYLLABUS**

**Unit 1**

**[15Hrs]**

**Software Testing principles**

Software testing principles, Levels of software testing, Test activities, SDLC and Testing, Verification & Validation, Quality Assurance, Quality Control.

## **White Box Testing Techniques**

Statement coverage, Branch Coverage, Condition coverage, Decision/Condition coverage , Multiple condition coverage, Inspections, Walkthroughs Code Review.

## **Black Box Testing**

Boundary value analysis, Equivalence partitioning, Cause Effect Graphing.

**Unit 2:**

**[20Hrs]**

## **Functional Testing**

Performance Testing, Stress testing, Configuration Testing, Security Testing, Recovery Testing, Integration Testing, Regression Testing, and Acceptance Testing.

**Unit 3:**

**[10Hrs]**

## **Testing process**

Comparison of different techniques, Test Plan, Test case Design Procedure Specification, Test Case Execution and Analysis, Test Documentation, Reporting test results

## **Testing web Application**

Testing concepts for web apps, Content Testing, User Interface Testing, Component Level Testing, Navigation Testing, Configuration Testing, Security Testing, and Performance Testing.

## **REFERENCES**

### **Mandatory Reading:**

1. Desikan, S., & Ramesh, G. (2006). *Software testing: principles and practice*. Pearson Education India.

### **Supplementary Reading:**

1. Kit E. *Software Testing in the Real World*, United States: Addison-Wesley Publishing Co.
2. Lewis, W. E. (2017). *Software testing and continuous quality improvement*. Auerbach publications.

**Web References :**

1. [www.guru99.com/software-testing.html](http://www.guru99.com/software-testing.html)
2. [https://www.tutorialspoint.com/software\\_testing/index.htm](https://www.tutorialspoint.com/software_testing/index.htm)
3. <https://www.javatpoint.com/software-testing-tutorial>

**Practical:** Software Testing

**Credit:** 3

**Duration:** 45hrs

**Marks:** 75

List of suggested PRACTICALS using any testing tool such as Selenium or equivalent:

- |                                     |      |
|-------------------------------------|------|
| 1. Planning Test Cases              | (2P) |
| 2. Generating Test Cases/Test Suite | (2P) |
| 3. Enhancing Tests                  | (3P) |
| 4. Debugging Tests                  | (2P) |
| 5. Running Tests                    | (2P) |
| 6. Analyzing Results                | (2P) |
| 7. Test Reporting                   | (2P) |

\*\*\*\*\*

**Course Title:** Design Analysis of Algorithms

**Course Code:** CSD-SK15

**Marks:** 75

**Credits:** 3

**Duration:** 45hrs

**Prerequisite Courses:** Nil

**Course Objectives:**

1. To study paradigms and approaches used to analyze and design algorithms and to appreciate the impact of algorithm design in practice.
2. To ensure that students understand how the worst-case time complexity of an algorithm is defined, how asymptotic notation is used to provide a rough classification of algorithms and compare with one another, and how there are still some problems for which it is unknown whether there exist an efficient algorithm, and how to design efficient algorithms.

**Course Outcomes:**

On successful completion of this course students will be able to:

CO1: Explain basic concepts related to the design and analysis of algorithms.

CO2: Describe divide-and-conquer paradigm, Dynamic Paradigm and Greedy Paradigm and explain when an algorithmic design situation calls for it

CO3: Explain the major graph algorithms and their analyses.

CO4: Analyze the performance of an Algorithm.

CO5: Choose appropriate algorithm and design technique for solving problem.

**SYLLABUS**

**Unit 1:**

**[15 Hrs]**

**Introduction**

What is an Algorithm, Rules for writing Algorithms, Properties of Algorithms, Framework for design and analysis of algorithms(RAM model of computation),Recursive Algorithms, Space and Time Complexity by Tabular method(Performance Analysis).

**Divide and Conquer**

Elements of Divide and Conquer Algorithms, QuickSort algorithm, Merge sort analysis, Strassen's algorithm for matrix multiplication, Analysis of Binary Search,The Maximum subarray Problem.

**Unit 2:****[15Hrs]****Dynamic programming**

General Method, caching v/s computation, Fibonacci numbers by recursion, Fibonacci numbers by caching, Fibonacci numbers by dynamic programming, Optimal Binary Search Tree, Rod Cutting Problem.

**Greedy algorithms**

Elements of greedy strategy, Activity-selection problem, Job sequencing with deadlines. Knapsack problem.

**Unit 3:****[15Hrs]****Graph Algorithms**

Elementary graph algorithms- Techniques for Graphs(Graph cycles, Topological sorting, maximum flow) Minimum spanning tree, growing a spanning tree, Kruskal and Prim algorithms.

**Backtracking**

General Method, 8-Queen's problem, sum of subsets, Graph coloring.

**References****Mandatory Reading:**

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT press.

**Supplementary Reading :**

Aho, A. V., & Hopcroft, J. E. (1974). *The design and analysis of computer algorithms*. Pearson Education India.

**Web References :**

1. [www.tutorialspoint.com/design\\_and\\_analysis\\_of\\_algorithms/index.htm](http://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm)
2. <https://www.javatpoint.com/daa-tutorial>



3. <https://www.guru99.com/design-analysis-algorithms-tutorial.html>

**Practical:** Design and Analysis of Algorithms

**Credit :** 3

**Duration:** 45hrs

**Marks :** 75

Practical Assignments are to be done using a Programming Language for the following :

Programs to implement

1. GCD of 2 numbers using Iterative and Recursive approach (1P)
2. QuickSort (1P)
3. Binary Search using Recursive approach (1P)
4. Fibonacci numbers using Dynamic Programming approach. (1P)
5. Activity Selection Problem using Dynamic Programming approach. (1P)
6. job sequencing with Deadlines. (1P)
7. Knapsack Problem (1P)
8. Rod Cutting Problem. (1P)
9. Optimal Binary Search Tree. (2P)
10. Graph using matrix and linked list (2P)
11. Graph Coloring Problem (2P)

\*\*\*\*\*

**Course Title:** Cloud Computing

**Course Code:** CSD-SK16

**Marks:** 75

**Credits:** 3

**Duration:** 45hrs

**Prerequisite Courses:** Nil

**Course Objectives:**

1. To make students understand the key elements of cloud computing.
2. To understand the difference between deploying applications on the cloud and the local infrastructure.

3. To understand various cloud service models.

### **Course Outcomes:**

On completion of the course students will be able to:

CO1: Explain the core concepts of the cloud computing paradigm.

CO2: Characterize the different cloud services ie. Infrastructure, Platform and Software as a Service (IaaS, PaaS, SaaS).

CO3: Deploy application in a production environment.

CO4: Host a cloud platform like Apache OwnStack and Owncloud

### **Syllabus:**

#### **Unit 1**

[15Hrs]

#### **Overview of Computing Paradigm**

Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing

#### **Introduction to cloud computing:**

Cloud Computing definition, History of Cloud Computing, How Cloud Computing Works, Benefits and challenges of cloud computing, Issues for Cloud Computing.

#### **Unit 2**

[15Hrs]

Comparison with traditional computing architecture (client/server), Cloud Computing Service Models, Deployment Models- Public cloud, Private cloud, Hybrid cloud and Community cloud, Key drivers to adopting cloud, Impact of cloud on users, Governance in the cloud.

#### **Infrastructure as a Service (IaaS)**

Introduction to IaaS: IaaS definition, Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Docker Container, CloudStack, Open Stack, Virtual Machine (VM). Resource Virtualization: Server, Storage, Network. Examples: Amazon EC2, Load balancing

#### **Unit 3**

[15Hrs]

#### **Platform as a Service (PaaS)**

Introduction to PaaS: What is PaaS, Service Oriented Architecture (SOA). Cloud Platform and Management: Computation, Storage, Examples: Google App Engine, Microsoft Azure, Salesforce.com.

#### **Software as a Service (SaaS)**

Introduction to SaaS, Web services, Web 2.0, Web OS, Introduction to MapReduce, Case Study on SaaS.

## References:-

### Mandatory Reading:

1. Hill, R., Hirsch, L., Lake, P., & Moshiri, S. (2012). *Guide to cloud computing: principles and practice*. Springer Science & Business Media.

### Supplementary Reading:

1. Furht, B., & Escalante, A. (2010). *Handbook of cloud computing* (Vol. 3). New York: Springer.

### Web References:

1. [https://www.tutorialspoint.com/cloud\\_computing/index.htm](https://www.tutorialspoint.com/cloud_computing/index.htm)
2. <https://www.guru99.com/cloud-computing-for-beginners.html>
3. <https://www.techopedia.com/definition/2/cloud-computing>

## Practical: Cloud Computing

**Credit:** 3

**Duration:** 45hrs

**Marks:** 75

1. Setup owncloud [2]
2. Cloud Server Management using VestaCP/Froxlor/ISPConfig3 [4]
  - a. DNS Setup
  - b. Email Configuration
  - c. Domain management
3. Apache CloudStack [4]
  - a. Installation
  - b. VPN
  - c. Working with Instances, Network, Storage
4. Container Management Using Docker [5]
  - a. Installing docker
  - b. Creating containers
  - c. Package and run a custom app using docker

**Course Title:** Content Management System

**Course Code:**

**Marks:** 75

**Credits:** 03

**Course Prerequisite:** Nil

**Course Objectives:**

1. Use and manage different content management system.
2. Design and deploy websites developed using a content management system.

**Course Outcomes:**

On successful completion of this Course students will be able to:

CO1: Install and maintain content management systems

CO2: Develop website using different content management systems

CO3: Make use of plugins to add more functionality

CO4: Create users and manage them

**UNIT 1:**

**15 hours]**

**Wordpress**

WordPress dashboard, Types of users, WordPress settings panel, Permalinks and RSS feeds, Creating and managing posts, Setting up post categories, Creating and managing pages, Managing comments, Installing and updating plugins, Customising WordPress themes, WordPress theme options, WordPress Security / backup / domain transfers, Migration From Different Platforms, Optimization of WordPress Website, SEO Plugin

**Woocommerce**

Introduction to Woocommerce, Woocommerce installation, Creating product: Creating your product - General data, Inventory data, Shipping data, Attributes, Advanced data, Grouped products, Virtual products, Downloadable products, External/Affiliate products, Setting up categories, tags, and product images,

**UNIT 2**

**[15 hours]**

**Jomla**

Joomla Global Configuration, Article Manager, Archive Manager, Frontpage Manager, Section Manager, Category Manager, Media Manager, Menu Manager, Component Manager, Content Manager, Extensions Manager, Module Manager, Plugin Manager, Template Manager,

Understanding the concept of Joomla Positions, Changing the layout structure by changing the module position, Understanding Basic Joomla Template, Customizing Joomla Template, Building Custom Joomla Template, Understanding templateDetails.xml File, Creating templateDetails.xml File using tmpl\_builder, Linking CSS, Linking Javascript, Understanding include, Displaying content in XHTML, Creating template, installation package, Creating Custom Forms, Changing the Form appearance using CSS.

### **UNIT 3**

**[15 hours]**

#### **Drupal**

Drupal Overview, Drupal Site Building, Introduction to Drush, Setting a New Site Title and Logo, Adding More Users, Assigning Roles and Permissions to Site Users, Creating a Blog, Working with Blocks, Working with Views, Changing Your Site's Theme, Installing New Add-on Modules, Working with the Drupal Docroot Directory, Creating a Basic Drupal Module, Adding JavaScript to Your Drupal Module.

#### **Moodle**

Course categories – an overview, Creating courses, Course requests, Managing courses in bulk, Forms of enrolment, User profiles, Standard user actions, Manual accounts, User authentication, Assigning roles, roles Capabilities, Customizing your front page, The Moodle editor, Module plugins,

### **REFERENCE**

#### **Mandatory Reading:**

1. Douglass, R. T., Little, M., & Smith, J. W. (2006). *Building online communities with Drupal, phpBB, and WordPress*. Apress.

#### **Supplementary Reading:**

1. Ravensbergen, R. (2015). *Building E-Commerce Solutions with WooCommerce*. Packt Publishing Ltd.
2. Barnett, J. (2015). *Drupal 8 for Absolute Beginners*. Apress.
3. Büchner, A. (2016). *Moodle 3 administration*. Packt Publishing Ltd.

#### **Web References:**

1. <https://www.tutorialspoint.com/wordpress/index.htm>
2. [https://docs.moodle.org/22/en/Moodle\\_video\\_tutorials](https://docs.moodle.org/22/en/Moodle_video_tutorials)
3. <https://www.tutorialspoint.com/drupal/index.htm>
4. <https://www.tutorialspoint.com/joomla/index.htm>

**Course Title:** Lab Content Management System

**Course Code:** COM- III.SD-SK9

**Marks:** 25

**Credits:** 03

1. Installing and creating a wordpress website [1P]
2. Installing themes and working with a wordpress editor [1P]
3. Plugins - Contact form and SEO [1P]
4. Creating a blog website using wordpress [1P]
5. Installing woocommerce plugin, add products to the site [1P]
6. Customizing products by setting the product attributes [1P]
7. Develop an ecommerce website [2P]
8. Installing drupal and building a website using a template [1P]
9. User management [1P]
10. Develop website using drupal [1P]
11. Installing moodle and setting up the home page [1P]
12. Managing courses in moodle [1P]
13. User management in moodle [1P]
14. Creating a website using moodle [1P]

**Course Title:** Network Security

**Course Code:** CSD-SK15

**Marks:** 75

**Credits:** 3

**Prerequisite Courses:** Nil

**Course Objectives:**

To understand the theory and concepts of Network Security

**Course outcome:**

At the end of the course students will be able to :

**CO1:** Gain Knowledge of the threats, vulnerabilities and system risks

**CO2:** Understand cryptography, ciphers and encryption algorithms

**CO3 :** Compare and contrast symmetric and asymmetric encryption systems

**CO4:** Know about viruses, Trojan horses, worms, program flaws and the defenses against them

**Syllabus :**

**UNIT 1**

**[15 hrs]**

**Concepts of Security & Classical Encryption Techniques**

Introduction, The need for security, Security Approaches, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Classical Encryption Techniques : Substitution techniques, Transposition techniques, Steganography.

**Design Principle of Block Cipher [**

Block Cipher Operation: Electronic Code Book, Cipher Block Chaining, Cipher Feedback, Output Feedback, Counter, Feistel Cipher, The Data Encryption Standard.

**Cryptography**

**i. Mathematical Tools [3Hrs]**

Introduction to Number Theory, Modular Arithmetic, Prime Numbers, Euler's Totient Function.

**UNIT 2**

**[15 hrs]**

**Public Key Cryptography**

Principles of Public Key Cryptosystems, The RSA Algorithm, Other Public key cryptosystems, Diffie Hellman Key Exchange.

**Cryptographic Hash Functions**

Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Hash Functions Based on Cipher Block Chaining, MD5 Message Digest Algorithm, Secure Hash Algorithm SHA 512.

### **Message Authentication Codes and Digital Signatures**

Message Authentication Requirements – Message Authentication Functions –Requirements for Security of MACs,MACs Based on Hash Functions, HMAC, MACs Based on Block Ciphers, Data Authentication Algorithm. Digital Signatures, Digital Signature Standard.

## **UNIT 3**

**[15 hrs]**

### **Key Management & Distribution And User Authentication**

Introduction, Digital Certificate, Private key Management, The PKIX Model, Public key cryptographic standards ,XML, PKI and security

### **Program Security**

Flaws, Malicious code: viruses, Trojan horses, worms, Program flaws: buffer overflows, time-of-check to time-of-use flaws, incomplete mediation.

### **Firewall and Virtual Private Network**

Introduction to network security techniques: IP Security, firewalls, virtual private networks.

### **Mandatory Reading :**

1. William Stallings, —Cryptography and Network Security – Principles and Practices, Prentice Hall of India, Fifth Edition

### **Supplimentary Reading**

1. KahateAtul, —Cryptography and Network Security, Tata McGraw-Hill.

### **Web References:**

<https://www.tutorialspoint.com/cryptography/index.htm>

[https://www.tutorialspoint.com/internet\\_technologies/firewall\\_security.htm](https://www.tutorialspoint.com/internet_technologies/firewall_security.htm)

[https://www.tutorialspoint.com/webservices/web\\_services\\_security.htm](https://www.tutorialspoint.com/webservices/web_services_security.htm)



**Lab : Network Security Credits : 3**

**Marks : 75**

List of Practicals

1. Implementation of Caesar Cipher (1P)
2. Implementation of One-Time Pad (1P)
3. Implementation of Playfair Cipher (1P)
4. Implementation of Vignere Cipher (1P)
5. Implementation of Hill Cipher (1P)
6. Implementation of Data Encryption Standard Algorithm (2P)
7. Implementation of Image Steganography (1P)
8. Implementation of RSA Algorithm (1P)
9. Implementation of Digital Signatures using RSA Algorithm (1P)
10. Mini Project/ Case Study (3P)