| | |
|---|---|
| Course Title:**DATA STRUCTURES(P)** | Course Code : **33A** |
| Semester : **III** | Course Group : **DSC-V** |
| Teaching Scheme in Hrs (L:T:P) : **0:0:4** | Credits :**4** |
| Map Code : **G (Practical-Programming)** | Total Contact Hours: **60** |
| CIA : **40 Marks** | SEE # : **60 Marks** |
| Programme: **BSc-CS/BCA/B.Sc(IT)** | **# - Semester End Exam** |

1. **Arrays & Recursion Concepts of Program Development in C**
   - **a.** Towers of Hanoi Puzzle,
   - **b.** Factorial, Fibonacci series using recursion
   - **c.** Sum of rows and columns and diagonals using 2D array
   - **d.** Matrix Operations(Program to perform matrix operations)
2. **Linked List:**
   - **a.** Create a Singly Linked
   - **b.** Perform the Insertion operation.
   - **c.** Perform the deletion operations
3. **Array & Linked List Implementation of Stack**
   - **a.** Create a Stack and perform the push and pop operations using array
   - **b.** Create a Stack and perform the push and pop operations using Linked list
   - **c.** Conversion of Infix to Postfix (To convert an infix expression into postfix expression)
4. **Array & Linked List Implementation of Queue.**
   - **a.** Create a queue and perform the insert and delete operations using array
   - **b.** Create a queue and perform the insert and delete operations using Linked list
5. **Trees**
   - **a.** Printing the level order data in reverse
   - **b.** Maximum element in a binary tree

**TEXT BOOK:**

**T1**.NarasimhaKarumanchi: " Data Structure and Algorithmic Thinking with Python", CareerMonk Publications,2016

(Unit 1 – Chapter 1,2, Unit 2 – Chapter 3, Unit 3 – Chapter 4,5, Unit 4 – Chapter 6, Unit 5 – Chapter 6,7)

**T2**.G A V Pai : "Data Structures and Algorithms: Concepts, Techniques and Applications" , McGraw Hill Education (India) Pvt Ltd.

NPTEL Course (National Programme on Technology Enhanced Learning) Joined initiative of the IITs and IISC.

**T3**."Programming, Data Structures and Algorithms in Python", Dr. MadhavanMukund, Professor and Dean of Studies,

https://onlinecourses.nptel.ac.in/noc16_cs11

**SEMESTER - III**
**MAJOR-THEORY**
**WEB BASICS**

**OBJECTIVES :**

- To study fundamental concepts and terms that is need for using HTML and CSS.
- To show how to enter, edit, test and validate HTML and CSS for the web pages of a website.
- To define the contents and structure for a web page and shows how to code the CSS that does basic formatting to the HTML content.
- To learn how to use the CSS box model for spacing, borders, and backgrounds alayout as well as to work with links and lists.

| HOURS / WEEK - 5 | | |
|---|---|---|
| HOURS / SEMESTER - 75 | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** |
| **10** | **5** | **60** |

nd to use CSS for page

**UNIT I (LECTURE HOURS: 12)**

**An Introduction to HTML and CSS**

**The HTML and CSS for a web page** - HTML document, DOCTYPE declaration

**A short history of the HTML and CSS standards.** - HTML elements, style sheet, rule set

**THE HTML SYNTAX**

**The basic structure of an HTML document** - HTML elements, root element

**How to code elements and tags** - two elements with opening and closing tags, two empty tags, correct & incorrect nesting of tags

**How to code Attributes** - opening tag with attributes, empty tag with attribute, Boolean attribute, coding rules

**How to code comments and whitespace** - An html Document with comments and white space

**The CSS syntax**

**How to code CSS rule sets and comments** - Rule set, selector, declaration, comments.

**How to code basic selectors** - Element type, id or class.

**HOW TO USE APTANA TO WORK WITH HTML AND CSS FILES**

**How to create a project** - Creating a project

**How to open an HTML file** - opening an HTML file within a project

**How to start a new HTML file** - start a new file from any template

**How to edit an HTML file** - Aptana with an auto completion list for an HTML file common coding errors

**How to open or start a CSS file** - Start a new CSS file for a project from any template

**How to edit a CSS file** - Aptana with an auto-completion list for a CSS file, common coding errors

**How to preview and run an HTML file** - show preview and run buttons

**UNIT II (LECTURE HOURS: 13)**

**HOW TO CODE THE HEAD SECTION**

**How to code the title element** - Speakers and luncheons

**How to link to a favicon** - The link element is used to link a custom icon

**How to include metadata** - a head section includes metadata, SEO guidelines

**HOW TO CODE TEXT ELEMENTS**

**How to code headings and paragraphs** - Block elements h1 to h6, p

**special blocks of text** - pre , blockquote , address

**How to code inline elements for formatting and identifying text** - Formatting text such as i, b, sub, sup, br and identifying content are abbr, cite, code etc.,

**Character and core attributes.** - character entities are & amp, & lt, & gt etc. attributes are id, class, title, lang.

**How to code links, list, and images**

**How to code absolute and relative URLs** - Root- relative path, document-relative path

**How to code links and lists** - href attribute for < a > element, link to a web page in same folder, link subfolder of the parent folder, link based root folder, link to another website, Elements created ordered and unordered lists < ul >, < ol >,< li >**How to include images** - Attributes of < img > element are src, alt, height, width. **An Introduction to CSS**

**Three ways to provide CSS styles for a web page** - External style sheet, embedded style sheet, inline styles

**How to specify measurements and colors**

**How to specify measurements** - Common units of measure such as px, pt, em, %, relative units of measure with afixed border.

**How to specify colors** - specify colors, with a RGB value, with a RGB value that uses hexadecimal numbers

**HOW TO WORK WITH CASCADING STYLE SHEETS**

**How the cascade rules work** - User rule set, identify rule as important, cascade order for applying CSS rule set.

**How to use the developer tools to inspect the styles that have been applied** - Modern browser provide developer tool to inspect elements and styles.

**HOW TO WORK WITH TEXT**

**How to set the font family and font size** - Generic font families, a font-family rule in the body element that is inherited by alldescendants, a font-family rule in a descendant that overrides the inherited font family.

**How to set the other properties for styling fonts** - syntax for the shorthand font property.

**How to set properties for formatting text** - Properties for indenting, aligning & decorating text, CSS that specifies a text indent & horizontal alignment.

**How to float an image so text flows around it** - Float property is used to float an image & set margin around it using margin property.

**UNIT III (LECTURE HOURS: 12)**

**An introduction to the box model**

**How the box model works** - Formula for calculating height & width of a box.

**A web page that illustrates the box model** - CSS adds for types of border to the elements in HTML

**HOW TO SIZE AND SPACE ELEMENTS**

**How to set heights and widths** - Properties for setting height & width, set the height & width of the content area, set maximum &minimum width & height.

**How to set margins and padding** - Margins - margin-top, margin-right, margin-bottom, margin-left. Padding - padding-top,padding-right, padding-bottom, padding-left.

**How to set borders and backgrounds**

**How to set borders** - Border - Borders, syntax, set side border, width of border, style & color. **How to set background colors and images** - Properties for setting background color & image**HOW TO CODE LISTS**

**How to code unordered and ordered lists** - Unordered list is displayed as bulleted list, list items can contain inline elements, ordered list is displayed as numbered list.

**How to code nested lists and description lists** - Nest lists by coding one list as an item for another list, description list, definition list.

**HOW TO FORMAT LISTS**

**How to change the bullets for an unordered list** - Properties for formatting unordered list, values for list-style-type property.

**How to change the numbering system for an ordered list** - Values for the list-style-type property of an ordered list.

**How to change the alignment of list items** - Margin & padding to control the indenting for the item in an ordered or unordered list.

**HOW TO CODE LINKS**

**How to link to another page** - Four attributes of the < a > element

**How to format links** - Common CSS pseudo-classes for formatting links, properties for removing underlines & borders.

**How to use a link to open a new browser window or tab** - Target attribute of the < a > element is set to open a newwindow.

**How to create and link to placeholders** - Placeholder or anchor use id attributes of the < a > tag.

**How to link to a media file** - Media formats & MIME types, plugins.

**How to create email, phone, and Skype links** - Link type: E-mail, prefix mailto:, phone: tel:, skype: skype prefix is used.

**UNIT IV (LECTURE HOURS: 11)**

**Basic skills for working with images**

**Types of images for the Web** - Jpeg, gif, png & description

**How to include an image on a page and to resize an image** - Attributed of the < img > tag, CSS properties for sizingan image.

**How to align an image vertically and to float an image** - Property for aligning images vertically: vertical-align, commonkeywords. Property for floating images: float, clear.

**ADVANCED SKILLS FOR WORKING WITH IMAGES**

**How to work with thumbnails** - To download an image faster & can use image editor for sizing an image.

**How to do image rollovers** - Image rollover to use background images

**How to create image maps** - image map attributes are usemap, name.

**Basic HTML skills for coding tables**

**An introduction to tables** - Rows, columns, data cells, header, footer.

**How to create a table** - Elements of coding table: table, tr, th, td.

**How to add a header and footer** - Thead, tbody & tfoot elements make easier to style a table with CSS.

**Basic CSS skills for formatting tables**

**How to use CSS properties to format a table** - Formatting table - Border-collapse, border-spacing, padding, text-align,vertical-align, tr, td & th elements

**OTHER SKILLS FOR WORKING WITH TABLES**

**How to merge cells in a column or row** - Merge cells - Colspan & rowspan

**How to provide for accessibility** - Accessibility - Caption, headers & scope.

**How to nest tables and control wrapping** - A table with another table nested within one of its cells , Content of the cellsin a table - wrap.

**UNIT V (LECTURE HOURS: 12)**

**HOW TO USE FORMS AND CONTROLS**

**How to create a form** - Form - name, action, method, and target. Controls, HTTP request, get method.

**How to use buttons** - Button - Type, value, src, alt, height & width.

**How to use text fields** - Text field - Text field, password field, hidden field. Type, value, maxlength, size, autofocus,placeholder

**How to use radio buttons and check boxes** - Radio button & check boxes - Type, value, checked

**How to use drop-down lists and list boxes** - The optgroup & option elements - Label, value, selected. List boxes -Size, multiple.

**How to use text areas and labels** - Textarea element - Rows, cols, wrap. Label element - for.

**How to group controls with fieldset and legend elements and to use a file upload control** - Group controls, legendelement coded within a field element, disabled attribute. File upload control - Accept, multiple

**Other skills for working with forms**

**How to align and format controls** - Series of controls and labels are align, text-align property, rule set coded for inputcontrol, :focus pseudo-class to change appearance of a control.

**How to set the tab order and assign access keys** - Tabindex, accesskey attributes for setting the tab order & access keys.

**AN INTRODUCTION TO MEDIA ON THE WEB**

**Common media types for video and audio** - Common media types for video & audio, media player, plugins.

**Video codecs** - H.264, Theora, VP8

**Audio codecs** - AAC, FLAC, MP3, Vorbis, WMA.

**Audio and video support in current browsers** - Audio & video types, mobile devices, MIME type.

**How to encode media** - Encoder - Miro Video Converter.

**HOW TO ADD AUDIO AND VIDEO TO A WEB PAGE**

**How to use the object and param elements** - Object - type, data, width, height, param - name, value.

**How to use the embed element** - Embed element - type, src, width, height.

**How to fall back to Flash for backward compatibility** - HTML for playing a video & audio that falls back to Flash

**Text Books :**

T1: Murach's HTML5 and CSS3 | Edition:3rd Edition | SHROFF PUBLICATION & DISTRIBUTERS PVT.LTD. | Anne Boehm AND Zak Ruvalcaba(2016)

**Reference Books :**

T2 : Brilliant HTML5 and CSS3 | Edition:2011 | Pearson | James A. Brannan AND Josh Hill( )

# Enterprise Java Beans

**OBJECTIVES:**

To enable the students to

Get a standard overview of EJB rationale and its architecture.

Create session beans and entity beans and know to deploy the application in a server.

Use of JDBC to access a SQL database

**PEDAGOGY:**

Teaching aids used are Black board, OHP, Projector, Demonstration, Analogy and Group Discussion.

| TOTAL HOURS/SEMESTER –90 | | |
|---|---|---|
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** |
| **12** | **6** | **72** |

**UNIT I (LECTURE HOURS: 14)**

**THE MOTIVATION FOR EJB , DIVIDE AND CONQUER AND COMPONENT ARCHITECTURES**

**Distributed System** - Standard multitier deployment, role of Application server

**Components** - Pricing Components, Examples of Different vendors

**Building large systems** - Things to consider Like RMI,
Load balancing **Application Server** - Set of interfaces
between components and application server

**INTRODUCING ENTERPRISE JAVABEANS , THE EJB ECOSYSTEM**

**Importance of Java** - 3 main reasons and its features

**EJB as a Business Solution** - Tasks of EJB, GUIs, EJB as back-end to web services

**The bean Provider, The application Assembler,** - Vendors of EJB, Tasks of Application assembler

**The EJB Deployer, The system administrator** - Challenges of EJB deployment, JMX, Responsibilities

**The container and Server Provider, The tool vendors, Roles in EJB** - Runtime environment,
IDEs, Parities of EJB and theirrequirements

**J2EE , ENTERPRISE BEANS, DISTRIBUTED OBJECTS AND DISTRIBUTED OBJECTS MIDDLEWARE**

**The J2EE technologies** - Java platforms, EEd, EJB, RMI, JNDI, JDBC

**Types of Beans** - Session bean, Entity bean , Message driven definition, Clients interaction

**The foundation for EJB** - Local/Remote transparency, Tchnologies

**Explicit Middleware** - Gained through APIs, Pseudo code, downsides of this approach **Implicit Middleware** - Gained through declarations, Tasks and values of this approach

**CONSTITUTES AN ENTERPRISE BEAN**

**Enterprise Bean class** - Interface for all classes (Source code)

**EJB Object** - Services, Steps to interact with EJB container, Remote interface

**The Home object** - Responsibilities of Home object, Home interface, Its source code

**Local interfaces** - Steps involved in creation, Its process, Preview of javax.ejb. EJBHome

**Deployment Descriptors, Vendor Specific files** - Requirements for deployment, Creating ejb-jar files, Terms associated.

## UNIT II (LECTUREHOURS: 14)

**DEVELOP AN EJB COMPONENT**

**Remote interface** - Order of operations, Simple java program
**Local interface** - Source code for implementation, HelloLocal.java
**Home interface** - Methods in Home, Steps to create, Java Source code
**Local Home interface Bean class** - Differences, Exceptions, Create(), Remove(), EJB Contexts,

**DEPLOYING A BEAN , UNDERSTANDING HOW TO CALL BEANS**

**The optional EJB client JAR file** - Steps for deployment, Vendor specific files, Ejb.jar file,
**Looking up a home project** - RMI-IIOP, Use JNDI, Hello client source code

**RUNNING THE SYSTEM , INTRODUCTION TO SESSION BEANS**

**Server side output & Client side output** - Methods to invoke, JNDI requirement, Running the client.java
**Implementing component interfaces** - Reason for component interface, Parameter passing
**Session bean lifetime** - Persistent objects, Non persistent objects,
**Session bean subtypes** - Stateful session bean, Stateless session bean, Its invocation, Method associated

**SPECIAL CHARACTERISTICS OF STATEFUL SESSION BEANS**

**Achieving the effect of Pooling with stateful beans** - Context Switching, Passivation, Virtual memory, Transaction
**The rules governing conversational state** - Object serialization, Code of session bean
**Activation and Passivation Callbacks** - Description of methods ejbActivate(), ejbPassivate(), Activation, Passivation process

**METHOD IMPLEMENTATION**

**A simple stateful session bean** - Count bean, Its required interfaces, Source code of interfaces, Client's code , I/O process
**Lifecycle diagrams for session bean.** - Steps involved, Methods required, diagrammatic representation.

## UNIT III (LECTURE HOURS: 14)

**ENTITY BEANS - PERSISTENCE CONCEPTS, ENTITY BEANS**

**Java object serialization** - RMI, JNDI, Querying objects, Example
**Object relational mapping** - Rational database, Example account database, Relationship to other data
**Object databases** - ODBMS, Object query language, Object database technology , Vendors

**About the files that make up an Entity bean** - Application , Persistent Logic component, bean class, Primary key class

**FEATURES OF ENTITY BEANS**

**Entity beans survive failures , Entity bean instances are a view into a database** - Difference between session and entity beans, Methods ejbLoad(), ejbStore(), Lading and storing process
**Several entity bean instances may represents the same underlying data** - Threads of execution, Handling simultaneous client requests, Thread safe instance, Performance boosting
**Entity bean instances can be pooled** - Decide the fields in database, Callback methods , role of EJB container
**Two ways to persist entity beans** - State save , State load, Persistence API, Container managed persistence

**Creation and removal of entity beans** - Understanding how to create and remove entity beans, Methods in all interfaces

**Entity beans can be found** - Data representation, Relationship between remove() and ejbremove()

**Modify entity bean data without using EJB** - Manual modification, using home object for all manipulation.

**ENTITY CONTEXTS**

**ENTITY BEAN CODING BASICS**

**getEJBLocalObject()/getEJBObject()** - Code of Context interface, Entitycontext interface

**getPrimaryKey()** - When to call the function, How to call the function, Methods associated to primarykey(),

**Finding Existing Entity beans ejbfind()** - Entity bean, Enterprise bean interface, Methods like, Setentitycontext(), ejbPostCreate(),ejbLoad()

**THE BEAN MANAGED PERSISTENT EXAMPLE**

**Creation of Home, Local interface** - Take bank example, Create object Model, Write source code of Home, Local interface, **Generate bean** - Source code of Account.java with Main methods, such as getBalance(), getAccount(), Business logic methods **Create Client program** - Get reference to interfaces, Accept the required details, Necessary method invocations

**DEPLOYMENT DESCRIPTOR, RUNNING THE CLINT PROGRAM**

**Setting up the database** - Create DDL for account table, Find JDBC driver and JNDI locations

**Server side output** - Obtain output based on EJB container behavior, finder method

**Client side output** - Produced result based on the input in client program and method invocations

**Bean's life cycle.** - Necessary methods for implementation of entity bean , Steps for entire process

**UNIT IV (LECTURE HOURS: 15)**

**FEATURES OF CMP ENTITY BEANS**

**CMP entity beans are Subclassed, No Declared fields** - Subclassing concept, Examples

**Get/Set methods are defined in the subclass, Abstract persistence schema** - Abstract Class, get, set methods coding, Deployment descriptor snippet

**Query language, ejbSelect() methods** - Finder methods, The process of developing and deploying a CMP bean

**IMPLEMENTATION GUIDELINES, CMP EXAMPLE**

**List of Methods and its typical implementation** - Methods such as ejbLoad(), ejbStore(), ejbRemove(), unsetEntitycontext()

**local, home, localhome interfaces** - Object model, Methods needed for home, Local interfaces **Container specific Deployment descriptor** - Use custom primary key classes, finally() classes

**DEPLOYMENT DESCRIPTOR OF CMP, MESSAGE DRIVEN BEANS**

**JAVA MESSAGE SERVICE**

**Running the program, Life cycle of CMP entity bean** - Run server side program, Give input for client program and obtain thenecessary results, Steps involved in life cycle

**Motivation to use of Message driven beans** - Asynchronous Programming, concern of messaging, RMI vs messaging **Messaging domainJMS API** - Pub/Sub, Point to point (PTP,)Client view of JMS sytem, Programming model, JMS interfaces, **INTEGRATING JMS WITH EJB, DEVELOPING MESSAGE DRIVEN BEANS**

**Approaches to integrate JMS with EJB** - Using java objects, Reuse existing type of EJB component, Characteristics of messagedriven beans

**The semantics** - Ejb.messagdrivenbeans(), Messagedrivencontext, necessary methods

**A simple example** - List of methods and their description, Implementation class, Life cycle, Client program

**ADVANCED CONCEPTS, MESSAGE DRIVEN BEAN GOTCHAS**

**Support Containers** - Transactions, Load balancing, Duplicate consumption in a cluster

**Message Ordering** - Message driven beans in a cluster, Queuing concept

**Missed ejbRemove() calls, Poison Messages** - How beans cause poison messages, Strategy to resolve,

**Return back to message producers** - A simple request/response paradigm solution, Advantage and disadvantage of using thisarchitecture,

**UNIT V: (LECTURE HOURS :15)**

**INTRODUCTION TO JDBC**

**Goals of JDBC** - Structure of JDBC, Basic class and interfaces, Databases and drivers, Alternatives to JDBC

**Connecting to the database** - Connection troubles, JDBC classes for connection creation

**Basic database access SQL dtatypes and Java datatypes** - Classes for access, SQL Null vs Java null, Clean up, Modifying the database SQL numeric, SQL long, JDBC prescribed sql datatypes

**Scrollable result sets** - Result set types, result set navigation, Driver with scrollable result set

**The JDBC support classes** - Java.sql.types, SQLexception, SQLwarning and Datatruncation

**A database servlet** - Getting configuration information, Random visitor comments, Saving, generating and inserting new comments

**ADVANCED JDBC**

**Prepared SQL** - Prepared statements, Stored Procedures,

**Batch processing** - Kind of statements to use, Efficient approach, Methods for batch processing **Updatable result sets** - Update, deletes, Inserts, Visibility of changes, Refreshing data from database**Advanced data types** - Blobs and Clobs,Arrays, SQL3 types, Type mapping

**Meta-data** - Result set, database meta-data,

**THE JDBC OPTIONAL PACKAGE**

**Data sources** - Naming and directory services, JDBC data sources, JNDI service provider **connection pooling** - Working Model , Regular connection API, Open, talk, close connection**Row sets** - Resultset interface, Swing application, Configuration, Usage, Row set events **Distributed transactions** - Data sources, Sample program

**OTHER ENTERPRISE APIS, STRUCTURE OF RMI, AN OBJECT SERVER, EJB ROLES, KINDS OF BEANS.**

**Java naming and directory interface** - Object binding, Object lookup

**Remote method invocation** - Structure of RMI, Remote interfaces, Object Server, Stubs and skeleton

**Object serialization** - String, Hashmap, Example SerialDemo

**Enterprise javabeans** - EJB roles, Kinds of beans, Delimits of RMI

**Text Books :**

1. Mastering Enterprise JavaBeans | Edition:2 | Wiley Student Edition | Ed Roman, AND Scott Ambler, AND Tyler Jewell( )

Database Programming with JDBC and Ja

**Programming Lab-IV ( Algorithms and Enterprise Java Beans )**
**B.Sc(CS)**

**OBJECTIVES:**

- Understand the time complexity of various data structures
- Selecting suitable data structures for any given problem
- Design Optimized algorithm and implement them for a given problem

**PEDAGOGY:**

- Teaching aids used are Projector, Demonstration and Group Discussion.

| TOTAL HOURS/WEEK - 3 | | | | |
|---|---|---|---|---|
| **TOTAL HOURS/SEMESTER - 45** | | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** | **TOTAL** |
| **6** | **3** | **36** | **-** | **36** |

## UNIT- I (LECTURE HOURS: 18)

1. **Array**

   Imagine a sorted array named "sarray" of integer type with size 12. Let the elements stored are: 100,99,88,77,66,55,44,33,22,11,10,9. Insert an element : 69.

   The constraint is that, the array should remain sorted even after the insert operation. Determine the worst case complexity of this process.

2. **Linked List**

   The linked list "LL" has the following elements stored in it. LL=11,22,33,44,55,66. Now try to insert 37 and analyse this operation for the worst case complexity .

3. **Array Comparision**

   Create two arrays of same type and store elements in both the arrays. Now try to find, out of these two arrays whether one array is the subset of an another array. State the complexity of your program.

4. **Heap**

   Heapify : 856,752,985,659,526 and determine the worst case complexity of your code.

5. **Sorting** Assume the given input: 45,85,74,45,12,65,10,6,8,36 and perform the insertion sort and find the worst case complexity of the sorting procedure.
6. **Search**

   Implement Sequential search and Binary search and state the complexities of both and reveal what sort is to be used when.

## Enterprise Java beans Lab

## Lab Experiments:

1. Stateless session bean to implement Banking operations

   Develop an application using stateless session bean for performing Banking operations such as Deposit, Withdraw and Find balance by accepting Account number and initial balance as input. Client should get output as menu driven and accept choice from user to produce result.

   **Sample Output:**

   java Client
   Before Handle
   Context successful
   Lookup successful
   Home stub obtained
   Enter the account number: 27345
   Enter Initial amount: 25000
   Successfully inserted
   The Menu
   1. Balance
   2. Credit
   3. Debit
   4. Exit
   Enter the Choice: 1
   Available balance: 25000.0

2. Stateful Session Bean for Library operations

   Deploy a project to carry out the book issue and return process for a library by using stateful session bean . A bean program which provide the implementation for the required methods. Client program should get the customer name and the various books he wants, also perform the other operations accordingly

**Sample Output:**

java Client

Enter details
Enter the A/C number: 27345
Enter the book title: C++ Programming
Enter the Author: Balagurusamy
Enter the no-of copies: 2
LIBRARY OPERATIONS
1. ISSUE
2. RETURN
3. EXIT
Enter ur choice: 1
Entering
Book Id is: 27345
Book Title: C++ Programming
Book Author: Balagurusamy
No-of copies: 1
Success

3. Entity Java bean for Payroll implementation

Develop and deploy an entity java bean which accepts employee details such as name of the employee, basic pay , deductions and allowances and display the payroll with net pay by applying the formula (Basic pay +Allowances-Deductions) in a neat format.

**Sample Output:**

java Client

Enter details
Name of Employee : A
Basic Pay : 25000
Deductions: 3000
Allowances: 5000
Net pay: 27000

4. Entity java bean for inventory

Implement an entity java bean for performing stock maintenance which exhibits purchase and sales details. Accept Stock as input and produce result as per the choice of the user.

**Sample Output:**

java Client

Enter details
Product name : xxx
Qty: 123
Price: 200
Inventory
1. Purchase

2. Sales
3. Stock

   Enter your choice: 1
   Product name: xxx
   Qty :20
   Stock : 143

5. Select records from Database

   Create an employee table with fields emp.id, Age, name in SQL and Create a JDBC application to Select/ fetch records from a table.

**Sample Output:**

Java JDBCselect

Connecting to a selected database…
Connected database successfully…
Creating statement….
ID :100, Age:25, Name : xxxx
ID :101, Age:27, Name : yyyy
ID :102, Age:23, Name : zzzz
ID :103, Age:22, Name : aaaa

6. Update operation in a database

   Create a JDBC Application to update age of employees as 30 whose id =100 , 101 in the existing employee table.

   **Sample Output:**

   Java JDBCupdate

   Connecting to a selected database…
   Connected database successfully…
   Creating statement….
   ID :100, Age:30, Name : xxxx
   ID :101, Age:30, Name : yyyy
   ID :102, Age:23, Name : zzzz
   ID :103, Age:22, Name : aaaa

**SEMESTER VI**
**MAJOR PAPER - XIII**
**JSP AND SERVLETS**

**OBJECTIVES:**

- To inculcate knowledge on JAVA servlet and Java Server Pages concepts.
- To understand the basic concepts of MVC patterns, sessions, cookies, client and server communication and data base
- Enable the students to apply the techniques of Servlet and JSP to develop web applications.

**PEDAGOGY:**

Teaching aids used are Black board, Projector, Demonstration, Analogy and Group Discussion.

| TOTAL HOURS/WEEK - 6 | | | |
|---|---|---|---|
| TOTAL HOURS/SEMESTER - 90 | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** |
| **12** | **6** | **72** | **-** |

**UNIT-I (LECTURE HOURS: 15)**

**An introduction to web applications**

**A typical web application and The components of a web application** - The first and second page of shopping cart application, components of a web application, description

**How static web pages work** - How a web server processes static web pages, description **How dynamic web pages work** - How a web server processes dynamic pages, description**AN**

**INTRODUCTION TO SERVLET/JSP WEB DEVELOPMENT**

**The software components** - The components of a servlet /JSP application, description

**The architecture** - The architecture for a typical servlet/JSP application, description

**The standard directories and files** - The directory structure for a web application named music store, A summary of thedirectories and files for a web application, description

**Two patterns for servlet/JSP applications**

**The Model 1 pattern, The Model 2 (MVC) pattern** - The Model 1 pattern, description - The Model 2(MVC) pattern,description

**A servlet/JSP application that uses the MVC pattern**

**The user interface and The HTML for the frst page** - The HTML page that gets data from the user, the JSP thatdisplays data, description - the index.html file, description

**The CSS for both web pages and The servlet for the back-end processing** - The main.css file in the styles subdirectory, description - The EmailListServlet class

**The web.xml file and The User class** - The web.xml file, description - The User class, description

**The JSP for the second page** - The thanks.jsp file , description

## UNIT-II (LECTURE HOURS: 15)

**How to work with HTML and JSP files**

**How to add an HTML or JSP file** - The new file dialog box, description **How to edit an HTML or JSP file** - The HTML and JSP editor , description**How to work with Java files**

**How to add a Java class and How to add a servlet** - The new java class dialog box, description - The first new servlet dialog box, The second dialog box, description

**How to edit a Java file** - The Java editor with an error displayed, description

**Other skills for working with web applications**

**How to add existing files to a project and How to deploy a web application to a remote server** - Figure 3-12 how to use the Files window

**How to work with a web application server and How to add a class library or a JAR file to a project** - Figure 3-13how to use the Services window - Figure 3-14 how to add a class library or a JAR file to a project

**How to register a database connection** - Figure 3-15 how to register a database connection

**How to code HTML forms**

**How to code a form and How to code text boxes** - Figure 4-13 how to code a form - Figure 4-14 how to code textboxes

**How to code buttons and How to code check boxes and radio buttons** - Figure 4-15 how to code buttons - Figure4-16 how to code check boxes and radio buttons

**How to code combo boxes and list boxes** - Figure 4-17 how to code combo boxes and list boxes

## UNIT-III (LECTURE HOURS: 14)

**How to request a servlet**

**How to use the HTTP GET method and How to use the HTTP POST method** - Figure 5-4 how to request a URL withthe HTTP GET method - Figure 5-5 how to request a URL with the HTTP POST method

**When to use the HTTP GET and POST methods** - get(read) and post(write)

**Skills for working with servlets**

**How to get the values of the parameters and How to get the real path for a file** - Figure 5-6 How to get the values ofthe parameters - Figure 5-7 How to get the real path for a file

**How to get and set request attributes** - Figure 5-8 How to get and set request attributes

**How to forward requests and How to redirect responses** - Figure 5-9 How to forward requests and How to redirect responses

**How to use JSP tags and How to include a file in a JSP**

**How to code directives, script lets, and expressions and How to code comments in a JSP** - Figure 6-4 How to use JSP tags - Figure 6-5 How to code comments in a JSP

**A JSP that includes a header and footer file and Three techniques for including files in a JSP** - Figure 6-9 A JSPthat includes a header and footer file - Figure 6-10 Three techniques for including files in a JSP

**UNIT-IV (LECTURE HOURS: 14)**

**An introduction to session tracking**

**Why session tracking is difficult with HTTP and How session tracking works in Java** - Figure 7-1 An introduction tosession tracking

**An application that needs session tracking** - Figure 7-2 An application that needs session tracking

**How to work with sessions**

**How to set and get session attributes and More methods of the session object** - Figure 7-3 How to set and get session attributes - Figure 7-4 More methods of the session object

**How to provide thread-safe access to the session object** - Figure 7-5 How to provide thread-safe access to thesession object

**How to work with cookies**

**An introduction to cookies** - Figure 7-6 An introduction to cookies

**How to create and use cookies** - Figure 7-7 How to create and use cookies **How to view and delete cookies** - Figure 7-8 How to view and delete cookies**The Cart application**

**The user interface and The code for the business classes** - Figure 9-11 The user interface for the Cart application - Figure 9-12 The code for the business classes

**The code for the servlets and JSPs** - Figure 9-13 The code for the servlets and JSPs

**UNIT-V (LECTURE HOURS: 14)**

**How to work with JDBC**

**An introduction to database drivers and How to connect to a database** - Figure 12-1 An introduction to databasedrivers - Figure 12-2 How to connect to a database

**How to return a result set and move the cursor through it and How to retrieve data from a result set** - Figure 12-3 How to return a result set and move the cursor through it - Figure 12-4 How to retrieve data from a result set

**How to insert, update, and delete data and How to work with prepared statements** - Figure 12-5 How to insert,update, and delete data - Figure 12-6 How to work with prepared statements

**The SQL Gateway application**

**The user interface and The code for the JSP** - Figure 12-7 The user interface for the SQL Gateway application - Figure12-8 The code for the JSP

**The code for the servlet and The code for the utility class** - Figure 12-9 The SQL Gateway servlet class - Figure12-10 The SQLUtil class

**The Email List application**

**The user interface and The code for the JSP** - Figure 12-13 The user interface and JSP for the Email List application **The code for the servlet and The code for the database class** - Figure 12-14 The code for the servlet for the Email List application -Figure 12-15 The database class for the Email List application

**The code for the utility class** - Figure 12-16 The utility class for the Email List application

**Text Books :**

1.Java Servlets and JSP | Edition:3 | Mike Murach & Associates | Joel Murach AND Michael Urban(2014)

**Reference Books :**

1. Web Technology and Application Management | Edition:1 | Wiley Publication | Hiren Joshi(2011)
2. Head First Servlets and JSP | Edition:2 | Shroff Publication | Basham AND Bates AND Bert AND Bryan AND Kathy AND Sierra(2008)

## SEMESTER – V

## MAJOR PAPER - XI

## PHP & My SQL

**OBJECTIVES:**

- Understand and demonstrate analytical powers of thought through critical analysis, evaluation and synthesis of fundamentals of PHP language and its associated Technologies.
- Demonstrate awareness on using PHP to create dynamic interactive Web forms.
- To create flexibility in designing, planning and building real-world projects.
- Develop specific focus and depth of critical understanding and interpretation in specific areas in the Web Development

**PEDAGOGY:**

- Teaching aids used are Black board, OHP, Projector, Demonstration and Group Discussion.

| TOTAL HOURS/WEEK – 5 | | | |
|---|---|---|---|
| TOTAL HOURS/SEMESTER – 75 | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** |
| **10** | **5** | **60** | **-** |

**UNIT I (LECTURE HOURS: 15)**

**WHAT IS PHP?**

**&**

**WHY USE PHP ?**

**A dynamic web page** - Interactive web site

**Process of running PHP script** - Clicking a link, requested URL, Script runs

**Common examples of PHP scripts** - Web forum, Online stores, Search enginees,Blogs,Web mail applications

**Feature of PHP** - Cross Platform, Integrate with all web servers

**How does PHP compare with other common Web programming technologies?** - Asp,Asp.Net,Perl,Java,Python,Ruby, ColdFusion

**120THE EVOLUTION OF PHP**

**&**

**WHAT'S NEW INPHP**

**5.3 Early history of PHP** - PHP version 2,PHP version 3.3,PHP 4,PHP 5

**Namespaces** - Avoid naming clashes

**The goto Operator** - Example

**Nowdoc Syntax** - Embed a block of PHP code

**Shorthand Form of the Ternary Operator** - Syntax
**Advanced Changes** - Late static bindings, Optional garbage collector
**INSTALLING PHP**
**Installing on Ubuntu Linux** - Steps to install on Ubuntu Linux **Installing on Windows** - Steps to install WampServer **Installing on Mac OSX** - Steps to install on Mac OS
**Testing Your Installation** - Testing the Web Server **Testing PHP** - Open text editor,type port number,phpinfo()
**Setting Your Time Zone** - Steps to set your time zone **OTHER WAYS TO RUN PHP**
**Running PHP with other Web Servers** - Internet Information Server, ISAPI module
**Compiling PHP Yourself** - Basic steps to compile PHP
**Running PHP Remotely** - Ssh package
**CREATING YOUR FIRST SCRIPT**
**Embedding PHP within HTML** - Code to create a Stylish Page, embed PHP within an HTML page
**Enhancing the Script Further** - Example to enhance the script to display the current time
**Using Comments to Make Code More Readable** - Single line comments and multi - line comments
**USING VARIABLES IN PHP & UNDERSTANDING DATA TYPES**
**Naming Variables** - Rules for naming variables, example
**Creating Variables** - Initializing a variable, example of declaring and initializing a variable
**About Loose Typing** - Loosely - typed language,example
**Testing the Type of a Variable** - gettype() function, PHP ' s type testing functions **Changing a Variable's Data Type** - settype() function,example **Changing Type by Casting** - List of casting functions that used in PHP
**OPERATORS AND EXPRESSIONS**
**Operator Types** - Arithmetic, assignment, logical, comparison, error control, incrementing-decrementing, Array, string

**Understanding Operator Precedence** - Precedence of Some PHP Operators (Highest First)
**Constant values** - define() function
**MAKING DECISIONS**
**Simple Decisions with the if statement** - if syntax and its example
**Providing an Alternative Choice with the else statement** - if…els, else…if syntax and its example
**Testing one Expression Many Times with the switch statement** - switch statement- syntax and its example
**Compact coding with the Ternary Operator** - ternary operator syntax with example
**180DOING REPETITIVE TASKS WITH LOOPING**
**Simple Looping with the while statement** - while syntax with example **Testing at the End: The do…while Loop** - do…while syntax with example **Neater Looping with the for statement** - The general syntax of a for loop
**Escaping from Loops with the break statement** - Explanation on break statement with an example **Skipping Loop iterations with the continue statement** - Explanation on continue statement with an example **Creating Nested Loops** - A Homing Pigeon Simulator:Application
**Mixing Decisions and Looping with HTML** - Fibonacci sequence example
**UNIT II (LECTURE HOURS: 15)**
**CREATING AND ACCESSING STRINGS**
**Including More Complex Expressions within Strings** - Common escape sequences
**Using Your Own Delimiters** - heredoc syntax and nowdoc syntax **Other ways to create Strings** - PHP function return string values **Finding the Length of a String** - strlen() , str_word_count() functions **Accessing Characters within a String** - About substr() function **SEARCHING AND REPLACING TEXT WITHIN STRINGS**

**Searching Strings with strstr()** - Syntax and usage of strstr() function with example
**Locating Text with strops() and strrpos()** - Syntax and usage of strops() and strrpos() function with example
**Finding the Number of Occurrences with substr_count()** - Syntax and usage of substr_count() function with example
**Searching for a Set of characters with strpbrk()** - Syntax and usage of strpbrk()function with example
**Replacing All Occurrences using str_replace()** - Syntax for str_replace(),substr_replace() and strstr() syntax with example **Replacing a Portion of a String with substr_replace()** - Syntax for str_replace(),substr_replace() and strstr() syntax with example **Translating Characters with strstr()** - Syntax for str_replace(),substr_replace() and strstr() syntax with example

**DEALING WITH UPPER- AND LOWERCASE:**
**FORMATTING STRINGS**

**String functions for converting upper and lower cases** - Syntax for strtolower(), strtoupper(), ucfirst(), lcfirst() functions
**General- Purpose Formatting with printf() and sprint()** - printf() versus sprintf(),Using type specifiers, Specifying Signs, Padding the Output, Specifying Number Precisions, Swapping arguments, Storing the Result Instead of Printing it
**Trimming Strings with trim(),ltrim() and rtrim()** - Purpose of trim functions and its syntax
**Padding Strings with str_pad()** - Adding space to the left and right of the string
**Wrapping Lines of Text with wordwrap()** - Splitting of lines to several lines using wordwrap() **Formatting Numbers with number_format()** - Purpose of number_format function and its syntax **CREATING & ACCESSING ARRAY ELEMENTS**
**Two types of array** - Indexed arrays , Associative arrays,how does PHP arrays work
**Creating a new array variable** - built-in array() construct
**Changing Elements** - Example
**Outputting an Entire Array with print_r()** - Explanation on print_r()
**Extracting a Range of Elements with array_slice()** - Explanation on array_slice()
**Counting Elements in an Array** - Explanation on count()
**Stepping Through an Array** - Manipulating pointer using current(), key(), next(), prev(), end() and reset() function and its descriptions

**LOOPING THROUGH ARRAYS WITH FOREACH**

**Using foreach to Loop Through values** - Syntax for - foreach… loop and a demo program to retrieve key and value pairs
**Using foreach to Loop Through Keys and Values** - Syntax for - foreach… loop and a demo program to retrieve key and value pairs
**Altering Array Values with foreach** - Syntax for - foreach… loop and a demo program to retrieve key and value pairs

**WORKING WITH MULTIDIMENSIONAL ARRAYS**

**Creating a Multi- dimenstional Array** - Demonstration of displaying an array of Books application **Accessing Elements of Multidimensional Arrays** - Demonstration of displaying an array of Books application **Looping through Multidimensional Arrays** - Demonstration of displaying an array of Books application **WHAT IS A FUNCTION? & WHY FUNCTIONS ARE USEFUL**
**Definition** - Subroutine,calling code
**Reasons to use functions** - Avoid duplicating code, easy to eliminate errors,reuseablitity
**Calling&**
**Working with Variable Functions**

**A function call** - Syntax for calling function and example
**WRITING YOUR OWN FUNCTIONS**
**Defining Parameters** - Use of function with defined parameters and default values
**Optional Parameters and Default values** - Use of function with defined parameters and default values
**Returning Values from your Functions** - Understanding local and global variable
**Understanding Variable Scope** - Understanding local and global variable.
**Writing Recursive Functions** - Definition,base case,recursive case,overview of recursive function, Creating theFibonacci Sequence with Recursion

**UNIT III (LECTURE HOURS: 14)**
**How HTMLForms Work**
**Form tag with its attributes** - Action,method,example, How it works, form fields, form elements with get and postmethod
**Capturing Form Data with PHP**
**Superglobal arrays** - $_GET, $_POST, $_REQUEST
**Dealing with Multi-Value Fields &**
**Generating Web Forms with PHP**
**Adding multi-value fields in php script** - adding square brackets ( [] ),example, two common approaches to generate a form within PHP
**Saving State with Query Strings**
**Building & Accessing Data Query Strings** - Query string,query string characters to be used within field names andvalues, http_build_ query() function, $_GET superglobal array
**WORKING WITH COOKIES**
**Cookie Components** - What is cookies,example to create a cookie,fields in cookie
**Setting a Cookie in PHP** - setcookie()function,example to use setcookie()
**Accessing & Removing Cookies in Your Scripts** - $_COOKIE superglobal array,example
**USING PHP SESSIONS TO STORE DATA**
**Creating a Session** - session_ start() function
**Reading and Writing Session Data** - $_SESSION[]
**Destroying a Session** - session_destroy() function, $_SESSION array, session_name() **Passing Session IDs in Query Strings** - PHPSESSID field, session_id() function **Changing Session Behavior** - Directives in php.ini file
**Understanding Files and Directories**
**File & Directories** - Importance of files and directories
**OPENING AND CLOSING FILES**
**Opening a File with fopen()** - fopen()function usage and explanation **Closing a File with fclose()** - fclose() function usage and explanation **Reading and Writing to Files**
**Functions to read and write the files** - About fread(),fwrite(),fseek(),feof(),fgets(), fgetc()
**Working with Dates and Times**
**Getting the Current date and time** - date(), getdate(), setdate(), checkdate(), time(), mktime()
**CREATING AND MANIPULATING IMAGES**
**Creating a New image** - About imagecreate()

**Allocating colors** - imagecolorallocate(), imagecolorresolve(),imagejpeg(), imagegif(),imagepng(),imagesetpixel(),imageline(),imageline(), imageellipse(),imagerectangle(), imagepolygon(), imagearc() **Outputting Images** - imagecreatefromjpeg(),imagecreatefromgif(),imagecreatefrompng()

**Drawing in an image** - imagecreatefromjpeg(),imagecreatefromgif(),imagecreatefrompng()
**Opening an Existing Image** - imagecreatefromjpeg(),imagecreatefromgif(),imagecreatefrompng()

**UNIT IV (LECTURE HOURS: 14) HOW TO HANDLE EXCEPTIONS**

**How to create and throw exceptions** - The syntax for a try-catch statement, A try-catch statement that catches an exception object,
A try-catch statement that throws an exception object, A try-catch statement that catches two types of exception ,description

**How to use the try-catch statement** - The syntax for creating a new exceptions, The syntax for the throw an exception, A statement that causes an exception to be thrown, methods of exception objects

**SETTING UP MYSQL**

**Starting the MySQL Server** - The MySQL server and its command-line tool, Ubuntu-WAMP Server on Windows, MAMP on Mac OS X

**Setting Up the MySQL root Password** - About root directory, root password, setting privileges

**A QUICK PLAY WITH MYSQL**

**Creating a Table Adding Data to a Table**

**Reading Data from a Table Updating Data in a Table Deleting Data from a Table Deleting Tables and Databases**

**Creating a New Database** - Explaination on CREATE DATABASE command,use command,create table,show tables, Insert,Update,Delete,Drop Commands and its examples

**CONNECTING TO MYSQL FROM PHP**

**Two main ways to connect** - Use of mysqli- and PDO (Data Objects)

**Making a Connection** - Setting with localhost and database name

**Reading Data** - Use of PDO Data object to fetch the data. Fruit shop application -Demo

**Setting Up the Book Club Database**

**Creating the Book Club Database** - Use of UNIQUE Constraint in Book club database for creating database andinserting records

**RETRIEVING DATA WITH SELECT**

**Limit the number of rows returned** - Use of select query and its various types **Sorting results** - Sort returned rows in any order with orderby query **Summarizing Data** - Use of count(), sum(), min(), max(), avg()

**Eliminate duplicate results** - Use of Distinct keyword in query

**Grouping Results** - Use of Groupby keyword in query

**Pulling Data from Multiple Tables** - Use of distinct keyword for combining tables

**UNIT V (LECTURE HOURS: 14)**

**INSERTING , UPDATING & DELETING RECORDS**

**Inserting a row of data** - INSERT query to insert a record

**Altering the data within an existing table** - UPDATE query to alter a record

**Deleting rows of data** - DELETE query to delete a record

**BUILDING A MEMBER REGISTRATION APPLICATION**

**Adding more Common code** - Defining validateField(), setChecked()

**Creating the Registration Script** - Creating the register.php file with internal .css and .html file.

**Testing the Application** - Explanation on register.php script

**What Is a Regular Expression?**
**Syntax for searching for patterns of text within strings** - Use delimiters(/),caret (^) character , strstr()function,example
**Pattern Matching in PHP**
**pattern - matching function** - preg_match(),function arguments, searching a particular position in the target string
**EXPLORING REGULAR EXPRESSION SYNTAX**
**Matching Literal Characters** - Special characters for matching literal characters, escape sequences
**Matching Types of Characters using Character Classes** - Shorthand character classes,example for character class, Dot to match any character
**Matching Multiple Characters** - Quantifiers ,example for matching a string of at least one digit
**Greedy and Non - Greedy Matching** - To match the largest number of characters, to match the smallest number of characters
**Using Subpatterns to Group Patterns** - preg_match(),$matches with example
**Referring to Previous Subpattern Matches** - Backreferences, example for referring to previous subpattern matches
**Matching Alternative Patterns** - Explanation on single and double vertical bar operator
**Using Anchors to Match at Specified Positions** - Assertion, caret ( ^ ) symbol, non - word character ( ! ), list of the anchors
**Finding Multiple Matches with preg_match_all()**
**Finding all matches for a regular expression** - preg_match_all() function,example for preg_match_all() function
**Searching Arrays with preg_grep()**
**Searching an entire array of strings** - Arguments of preg_grep(),example for preg_grep() function
**REPLACING TEXT**
**Replacing Text with preg_replace()** - Function str_replace(),Arguments of str_replace()
**Replacing Text using a Callback Function** - preg_replace_callback(),example to do arithmetic in regular expressions

**Text Books :**
Beginning PHP 5.3 | Edition:Wiley India Edition | Wrox Publication | Matt Doyle(2012)

**Reference Books :**
Beginning PHP and MySQL | Edition:Third Edition | From Novice to Professional | W. Jason Gilmore(2008)
PHP and MySQL Web Development | Edition:Fourth Edition | Pearson Publication | Laura Thomson AND Luke Welling(2011) PHP,Apache,MySQL® Web Development | Edition: | Wiley Publishing,Inc | Elizabeth Naramore AND Gary Mailer AND Jason Gerner ANDJeremy Stolz AND Michael Glass AND Yann Le Scouarnec(2004)

**SEMESTER VI**
**MAJOR PRACTICAL - VI**
**PROGRAMMING LAB –VI (JSP AND SERVLETS)**

**OBJECTIVES:**

- To apply knowledge of JAVA servlet and Java Server Pages for creating web applications.
- Enable the students to apply the techniques of sessions and cookies to develop web applications.
- To apply data base skills to web based applications

**PEDAGOGY:**

Teaching aids used are Projector, Demonstration, Analogy and Group Discussion.

| **TOTAL HOURS/WEEK – 3** |
|---|
| **TOTAL HOURS/SEMESTER – 45** |

**Use styles** - Algorithm and Coding.
**Modify a theme** - Algorithm and Coding.
**WORK WITH TABS.**
**Review the app** - Algorithm and Coding.
**Use a string to display the content for a tab** - Algorithm and Coding.
**Use a simple adapter to display the content for a tab** - Algorithm and Coding.

| CIA | CO-CURRICULAR ACTIVITIES | PRACTICAL | TUTORIAL |
|---|---|---|---|
| 16 | 3 | 36 | - |

**How to Structure a web application**
**Install and Use NetBeans** - Aim, Algorithm and Web Page(s) creation
**Experiment with an Existing Application** - Aim, Algorithm and Web Page(s) creation
**Working with Servlets and Web Applications**
**Modify the servlet for the Email list application** - Aim, Algorithm and Web Page(s) creation
**Create a new servlet** - Aim, Algorithm and Web Page(s) creation
**How to Develop JSP pages**
**Modify the JSPs for the Email list application** - Aim, Algorithm and Web Page(s) creation
**Create a new JSP** - Aim, Algorithm and Web Page(s) creation
**How to use session in web application**
**Use a session attribute** - Aim, Algorithm and Web Page(s) creation
**Use URL rewriting instead of hidden fields** - Aim, Algorithm and Web Page(s) creation
**How to work with JDBC**
**Create a user admin application** - Aim, Algorithm and Web Page(s) creation

TEXT BOOK:

**T1.** Java Servlets and JSP by Joel Murach(Author) and Michael Urban(Author),

Publisher: Mike Murach & Associates, Third Edition- 2014

Unit I: (Chapters 1 & 2) Unit II: (Chapters 3 & 4), Unit III: (Chapters 5 & 6)

Unit IV: (Chapters 7 & 9) Unit V: (Chapters 12).

REFERENCES:

**R1.** "Web Technology and Application Management" by Hiren Joshi (Author)
Publisher: Wiley, First Edition – 2011

**R2.** "Head First Servlets and JSP" by Basham (Author), Bryan (Author), Sierra (Author), Kathy (Author), Bates (Author), Bert (Author) ; Publisher: Shroff, Second edition -2008

**SEMESTER - VI**
**MAJOR-PRACTICAL**
**PROGRAMMING LAB - VI ( ANDROID PROGRAMMING AND MONGO DB )**

**DEVELOP A NEW APP.**
**Create the app** - Algorithm and Coding.
**Modify the app** - Algorithm and Coding.
**Introduce and correct a compile time error** - Algorithm and Coding.
**INVOICE TOTAL APP.**
**Create the project** - Algorithm and Coding.
**Create the user interface** - Algorithm and Coding.
**Write the Java code** - Algorithm and Coding.
**MODIFY THE LAYOUT FOR THE TIP CALCULATOR APP.**
**Test the relative layout version of the app** - Algorithm and Coding.
**Use a table layout** - Algorithm and Coding.

**UNIT V (LECTURE HOURS: 18)**
**ADD RADIO BUTTONS AND A SPINNER.**
**Test the app** - Algorithm and Coding.
**Add radio buttons** - Algorithm and Coding.
**Add a spinner** - Algorithm and Coding.
**USE BUILT-IN THEMES, USE STYLES, AND MODIFY A THEME.**
**Use built-in themes** - Algorithm and Coding.

## C PROGRAMMING

**OBJECTIVES:**

- To inculcate knowledge on C Programming concepts.
- To understand the basic concepts of Structured Programming Model, Identifiers, Arrays, Control Statements, Functions, Pointers, Formatted I/O, Characters and Strings, Structure, Unions and File I/O.
- Enable the students to apply the techniques of functions to develop console applications.

**PEDAGOGY:**

Teaching aids used are Black board, OHP, Projector, Demonstration, Analogy and Group Discussion.

| TOTAL HOURS/SEMESTER - 90 | | | |
|---|---|---|---|
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** |
| **12** | **6** | **60** | **12** |

**UNIT- I** **LECTURE HOURS :14**

**Introduction: Computers and Programming :** Moore's Law Computer organization machine languages, assembly languages, high level languages, interpreter Built for Performance, Standardization, The new C standard (computer programs, programmers, software, Moore's law logical units of a computer(table) machine, assembly and high level languages, interpreter portable, some popular performance oriented C Applications (table), hardware platforms) - Phase 1-6, standard input-output and error streams comments #include preprocessor directive blank lines and white space the main function an output statement escape sequences (typical C development environment(diagram) comments, document programs, multi-line comments, %...% multi line comments C preprocessor, standard input output header(<stdio.h>) white space function, body, left and right brace action, character string or message or literal, argument, statement, statement terminator escape character, escape sequence, new line, some common escape sequences(table)) - variable and variable definitions identifiers and case sensitivity syntax error prompting messages the scanf function and formatted inputs assignment Statement(Definitions, variables, integer, Example code: addition program identifier, case sensitive placing variable definitions among executables statements cause syntax error prompt message using printf with new line scanf, format control string, %d conversion specifier, address operator, Enter key, Interactive computing assignment statement, operands) **Memory, arithmetic and decision making:** Introduction: memory concepts integer division and the remainder operator arithmetic expressions in straight line form(type, value, memory location showing the name and value of a variable, destructive, memory location after both variables are input,

non destructive, memory location after a calculation arithmetic operators(table), integer division, remainder operator straight line form) - parentheses for grouping sub expressions rules of operator precedence sample algebraic and Cexpressions (Parentheses are used in the same manner in algebraic expressions four rules for operator precedence, associativity, precedence of arithmetic operators(table) expression calculates arithmetic mean(average), equation of a straight line, expression contains remainder(%)).Introduction: Decision Making (decisions, if statement, condition, equality and relational operators (table), Example code: using if statements, relational and equality operators , Precedence and associativity of the operators (table), C's keywords (table)) - **Structured program development :** Introduction: Algorithms and Pseudo code (Procedure: actions and order, Example: rise and shine algorithm, action statements) - Introduction : Control Structures Flow charts (Sequential execution, transfer of control, go to statement, control structures: sequence, selection and repetition structure special purpose symbols: rectangles, diamonds, rounded rectangles and small circles, Flow charting C's sequence structure(diagram), small circle symbol(connector), decision symbol(diamond)) - Selection Statements in C Repetition Statements in C (single selection statement, double selection statement, multiple selection statement single entry / single exit control statements, control statement stacking ) - **Assignment, Increment and Decrement Operators:** Several Assignment Operators Unary Increment and Decrement Operators (Additional assignment operator - Arithmetic assignment operators (table) increment operator, decrement operator , pre increment or pre decrement and post increment or post decrement, Increment and Decrement Operators (table), Example code: pre incrementing and post incrementing, Precedence and Associativity of the operators(table))

**UNIT – II** **LECTURE HOURS: 16**

**Control Structures:** Counter controlled repetition Sentinel controlled repetition Counter Controlled Repetition requirements (definite repetition, control variable sentinel values are used to control repetition name, initial value, increment (or decrement), final value) - for Statement Header Components General Format of a for Statement Comma-Separated Lists of Expressions (Example code: Counter controlled repetition with the for statement for Statement Header Components (diagram) General format of for and while statements ) - switch Statement Details switch Statement Flowchart Ignoring Newline, Tab and Blank Characters in Input Constant Integral Expressions Notes on Integral Types (controlling expressions-case labels Flowcharting a Switch multiple selection Statement with breaks skip Newline-Tab - Blank Characters test Constant Integral Expressions several data types to represent integers) - do…while Statement Flowchart break Statement, continue Statement (Example code: Using the

do…while repetition statement, Flowcharting the do…while Statement Example code: Using the break statement in a for statement, Example code: Using the continue statement in a for statement.) – **Functions:** Introduction: Program Modules in C Introduction: Math Library Functions(Functions, C standard library, programmer defined functions, calling function or caller, called function, Hierarchical boss-function /worker function relationship(diagram) argument, Commonly used math library functions(table)) - Introduction: Functions main's return Type Function maximum ( local variables, parameters, software reusability, abstraction Example code: Creating and using a programmer, defined) - **Logical and Assignment operators:** Logical AND (&&) Operator Logical OR Operator logical negation (!) operator (Truth table for the logical AND (&&) operator Truth table for the logical OR operator Truth table for operator ! (logical negation)) - Introduction: Passing Arguments by Value and By Reference (function, format of function definition Example code: Finding the maximum of three integers pass-by-value, pass-by-reference, side effects) - **Storage Classes, Scope Rules:** Local Variables Static Storage Class Introduction: Scope rules (Storage class specifiers: auto, register, extern and static, storage duration, linkage, automatic storage duration, static storage duration, automatic variables extern and static, several types of identifiers with static storage duration scope of an identifiers, function scope).

**UNIT – III** **LECTURE HOURS :10**

**Introduction: Arrays:** Defining Arrays Defining an Array and Using a Loop to Initialize the Array's elements (position number, zeroth element, 12-element array(diagram), subscript, name, value, Operator precedence and associativity(table) ,) - Initializing an Array in a Definition with an Initializer List Difference Between Passing an Entire Array and Passing an Array Element(Array initializers, Example code: Initializing the elements of an array with an initializer list %p conversion specifier, scalars, const, Example code: Array name is the same as the address of the array's first element, Example code: Passing arrays and individual array elements to functions) - **Multidimensional Arrays:** Two-Dimensional Array Manipulations(multi dimensional arrays, tables, double subscripted array, m by n array, Double-subscripted array with three rows and four columns (diagram), Example code: Initializing multidimensional arrays, Example code: Double-subscripted array manipulations.) – **Pointers:** Declaring Pointers Initializing and Assigning Values to Pointers Pointer Representation in Memory (indirection, Directly and indirectly referencing a variable(diagram) NULL is a symbolic constant address operator, Graphical representation of a pointer pointing to an integer variable in memory (diagram)) - The Indirection (*) Operator Demonstrating the & and * Operators Pass-By-Value Pass-By-Reference (Representation of y and yPtr in memory (diagram), Indirection Operator or dereferencing operator

Example code: Using the & and * pointer operators, Operator precedence and associativity(table) Cube a variable using pass-by-value Cube a variable using pass-by-reference with a pointer argumentRepresentation of y and yPtr in memory (diagram), Indirection Operator or dereferencing operator Example code: Using the & and * pointer operators, Operator precedence and associativity(table) Cube a variable using pass-by-value Cube a variable using pass-by-reference with a pointer argument) - Determining the Sizes of the Standard Types, an Array and a Pointer Introduction: Pointer Expressions and Pointer Arithmetic (sizeof, Example code: Applying sizeof to an array name returns the number of bytes in the array, Example code: Using operator sizeof to determine standard data type sizes Array v and a pointer variable vPtr that points to v (diagram), The pointer vPtr after pointer arithmetic (diagram), pointer to void(void *)) - **Arrays of Pointers:** String Copying with Arrays and Pointers (offset, pointer/offset notation, pointer/subscript notation, pointer subscripting, Example code: Using subscripting and pointer notations with arrays, Example code: Copying a string using array notation and pointer notation) - Definition: Arrays of Pointers (Arrays of Pointers array of string or string array, Graphical representation of the suit array (diagram). )- Using Function Pointers to Create a Menu-Driven System (Pointers to a Function, passed to functions, returned from functions, stored in arrays, assigned to other function pointers, Example code: Demonstrating an array of pointers to functions)

**UNIT – IV** **LECTURE HOURS :10**

**Character and String Handling:** Introduction: Fundamentals of Strings and Characters Functions isdigit, isalpha, isalnum and isxdigit Functions islower- isupper-tolower and toupper (Character constants, character set, special characters, string literals or string constant, null character, string is a pointer Character-handling library (<ctype.h>) functions (table), Example code: Using functions isdigit-isalpha- isalnum and isxdigit Example code: Using functions islower-isupper- tolower and toupper) - Functions isspace-iscntrl-ispunct-isprint and isgraph Functions fgets and putchar Function getchar (Example code: Using Functions isspace-iscntrl-ispunct-isprint and isgraph Standard input/output library character and string functions (table), Example code: Using Functions fgets and putchar Example code: Using Function getchar) - Functions strcpy and strncpy Functions strcat and strncat (String-manipulation functions of the string-handling library(table), Example code: Using functions strcpy and strncpy Example code: Using Functions strcat and strncat) - Introduction: Comparison Functions of the String-Handling Library (String-comparison functions of the string-handling library(table), Example code: Using functions strcmp and strncmp) – **Structures:** Self-Referential Structures Defining variables of Structure Types (derived data types, struct, structure tag, structure type and members, structemployee2- Self-

Referential Structures struct card) - Structure Tag Names Operations that can be performed on structures (Structure Tag Name is optional assigning structure variables to structure to structure variables of the same type, taking the address(&) of a structure variable, accessing the members of a structure variable and using the sizeofopearator to determine the size of a structure variable ) - Introduction: Initializing Structures Introduction: Accessing Structure Members (struct card aCard = { "Three", "Hearts" }; structure member operator(.) and the structure pointer operator (->), Example code: Structure member operator and structure pointer operator) - Introduction: Using Structures with Functions (Structures may be passed to functions by passing individual structure members, by passing an entire structure or by passing a pointer to a structure.) - Introduction: typedef (typedefstruct card Card).

**UNIT – V** **LECTURE HOURS :10**

**Unions and Enumeration:** Union Declarations Operations That Can Be Performed on Unions Initializing Unions in Declarations (union, share same space storage, union definition four operations can be performed on unions Union may be initialized with a value of the same type as the first union member) - Demonstrating Unions Introduction: Enumeration Constants (Example code: Displaying the value of a union in both member data types enumeration constants, Example code: Using an enumeration) - **Files and Streams**: Introduction: Files and Streams (end-of-file marker, stream, standard input-standard output-standard error-file descriptor, open file table, file control block (FCB), stdin-stdout-stderr, fgetc()-fputc()-fgets()- fputs(), fscanf()-fprintf(), fread()- fwrite().) - **Creating and Reading Sequential File :** Introduction: Creating a Sequential Access File (Example code: Creating a Sequential-Access File, End-of-file key combinations for various popular operating systems(table), Relationship between FILE pointers, FILE structures and FCBs(diagram), File opening modes(table)) - Resetting the File Position Pointer (Example code: Reading and printing a sequential file, file position pointer, file offset) - **Command Line Arguments:** Introduction (argc and argv, Example Code: Using Command Line Arguments).

**TEXT BOOK:**

**T1.** C - How to program, Paul and Harvey Deitel, 7th Edition, Pearson Education-Inc., 2013.

**REFERENCES:**

**R1**.The C Programming Language Edition: 2nd Prentice Hall Software Series Brian W. Kernighan ,DennisM.Ritchie(1988)

**R2.**Programming in C Edition:2nd Pearson Education Ashok N. Kamthane(2011)

**R3.** https://onlinecourses.nptel.ac.in/noc15_cs15/preview, Year 2015, Course Instructor: Dr.SatyadevNandakumar, Department of Computer Science and Engineering I.I.T. Kanpur and Prof.AmeyKarkare, Department of Computer Science and Engineering I.I.T. Kanpur

**R4.**http://www.geeksforgeeks.org/c/

**SEMESTER II**

| | | | |
|---|---|---|---|
| Course Title | : **INTRODUCTION TO COMPUTING USING PYTHON (T)** | CourseCode | : **23A** |
| Semester | : **II** | CourseGroup | : **DSC- III** |
| Teaching Scheme in Hrs (L:T:P) | : **4:0:0** | Credits | : **4** |
| Map Code | : **E(Theory-Programming)** | Total Contact Hours | : **60** |
| CIA | : **25 Marks** | SEE | : **75Marks** |
| Programme | : **BSC-CA /IT/CS** | | **# - Semester End Exam** |

| No. | Course Outcome | PSOs | Cl. Ses | CL |
|---|---|---|---|---|
| CO1 | Understand the FundamentalsofComputing, Programming andDebugging | PSO1 & PSO4 | 7 | U/R |
| CO2 | Define the Procedural Programming, Variables, Logical and Mathematical Operators | PSO1 & PSO4 | 7 | U/AP |
| CO3 | Determine the Control Structures, Conditionals and Loops. | PSO1 & PSO4 | 8 | AP |
| CO4 | Define the Functions and Error Handling | PSO1 & PSO4 | 8 | U/AP |
| CO5 | Explain the Data Structures, Strings and List | PSO1 & PSO4 | 10 | U |
| CO6 | Understand the File Input Output, Dictionaries | PSO1 & PSO4 | 10 | U/AP |
| CO7 | Design programs that uses Object-Oriented Concepts | PSO1 & PSO4 | 10 | AP |

**UNIT : I** **LECTURE HOURS: 14**

Computing: What Is Computing? ( Introduction to Programming, Programming Is Everywhere)-Programming Vocabulary ( Programs and Code, Input and Output, Compiling and Executing)- Programming Languages, Console vs. GUI ( The Console, GUIs)- Introduction to Python, Setting Up ( Python:

A High-Level Language, Python: An Interpreted Language, Files and the Command Line, Using an IDE, Web-Based IDEs, Interactive Mode). Programming: What Is Programming? & Writing Code: Lines ( The Programming Flow, Chaining Together Instructions, The Print Statement, Work in Small Chunks)- Writing Code: Lines in Python & Running Code: Compiling vs. Executing ( Your First Program: Hello, World, Printing Other Values, The Programming Flow, Compiling, Executing)- Executing Code in Python & Evaluating Results ( Encountering Errors, "Compiling" Python, The Python Interactive Mode, Errors, Incorrect Results). DEBUGGING: What Is Debugging? - Debugging in the Programming Flow Types of Errors & Types of Errors in Python ( Compilation Errors, Runtime Errors, NameError, TypeError, AttributeError, SyntaxError)- Basic Debugging & Basic Debugging in Python ( Print Debugging, Scope Debugging, Rubber Duck Debugging, Print Debugging in Python, Scope Debugging in Python).

**UNIT : II** **LECTURE HOURS: 16**

Procedural Programming: What Is Procedural Programming? & Procedural Programming in Python ( Functional Programming, Object-Oriented Programming, Event-Driven Programming, Hello, World, Data Types and Variables, Logical Operators, Mathematical Operators)- Comments and Documentation & Comments and Documentation in Python ( Comments, Self-Documenting Code, In-Line Comments, Code Block Comments, Self-Documenting Code) Variables: What Is a Variable? & Variables in Python ( Examples of Variables, Different Kinds of Variables, Python and Typing, Naming Rules and Conventions)- Assigning Variables & Assigning Variables in Python ( Give Values to Variables, Assign Values before Using Variables, Giving Variables to Values (Wrong!) )- Data Types & Data Types in Python ( Basic Data Types, Importance of Data Types, Common Types, The type() Function, Mixing Types)- Type Conversions in Python & Reserved Keywords in Python ( Converting to Strings, Converting from Strings, User Input, Python's Reserved Words, Misusing Reserved Words, What about functions?) Logical Operators: What Are Logical Operators? & Relational Operators ( Mathematical Operators,

Logical Operators, Numeric Comparisons, Non-Numeric Equality Comparisons, Set Operators)- Relational Operators in Python ( Numeric Equality Comparisons, Numeric Value Comparisons, Non-Numeric Equality Comparisons, Set Operators, Operators in Action)- Boolean Operators & Boolean Operators in Python , And, Or, Not, Combining Boolean Operators, Simplifying Conditionals) Mathematical Operators: What Are Mathematical Operators? & Mathematical Operators in Python (The Assignment Operator, Mathematical Operators, Additional Operators, The Basic Mathematical Operators, Modulus, Other Operators)- Self-Assignment and Incrementing & Self-Assignment and Incrementing in Python ( Self-Assignment, Incrementing, Self-Assignment, Self-Assignment Shortcuts, Incrementing and Loops)

**UNIT : III** **LECTURE HOURS: 10**

Control Structures: What Are Control Structures? & The Control Structures ( What Do Control Structures Do?, Conditionals, Loops, Function, Exception Handling)- Indentation and Control Structures in Python ( Indentation and Conditionals, Nested Indentation)- Scope & Scope in Python ( Examples of Scope, Control Structures and Scope, Simple Scope in Python, The Dangers of Scope in Python). Conditionals: What Are Conditionals? & Conditionals in Python ( If-Then, If-Then-Else, If-Then-Else-If, Multiple Else-Ifs, Conditionals Recap, Common Errors)- Conditionals and Operators & Conditionals and Operators in Python ( Relational and Mathematical Operators, Boolean Functions, Boolean Operators, Set Membership Operators). Loops: What Is a Loop? & Traditional For Loops in Python ( For Loops, While Loops, For Loops with Known Ranges, For Loops with Unknown Ranges)- For-Each Loops in Python & While Loops in Python (For-Each and Lists, For-Each and Other Types, Simple While Loops, While Loops and Number Guessing, Infinite Loops). FUNCTIONS: What Is a Function? ( Power of Functions, Function Terminology: Calls and Definitions, Parts of a Function Definition, Parts of a Function Call )-An Analogy for Functions ( Setting Up the Analogy, The Function Definition, The Function Call, Bigger Functions)- Simple Functions in Python & Functions with Returns and Parameters in Python ( The Function Definition, The Function Call, A Function with a Return, A Function with a Parameter, A Function with Multiple Parameters). Error Handling: What Is Exception Handling? & Try-Catch-Finally ( “Catching” Errors, When to Catch Errors, The Try Block, The Catch Block, The Finally

Block)- Try and Except in Python ( The Try Statement, Catching Any Error, Catching a Specific Error, Catching Multiple Specific Errors)- Else and Finally in Python ( Else for Error Handling, Else and File Input, Finally, Finally and Uncaught Errors, Nested Try-Catch-Else-Finally)

**UNIT : IV** **LECTURE HOURS: 10**

Data Structures: What Are Data Structures? (Advanced Data Types, Lists and List-Like Structures, Unit Outline)- Passing by Value vs. Passing by Reference & Passing by Value and Reference in ( Passing by Value: An Analogy, Passing by Reference: An Analogy, Terminology: By Reference, Integers: By Value or by Reference?, Other Data Types: By Value or by Reference?, Variable Assignments)- Mutability in Python ( Mutability vs. Passing by Reference, Reassigning Immutable Data Types, Immutable Data Types: Functions vs. Local Assignments, Printing Memory Addresses)- A Brief Introduction to Methods ( Functions vs. Methods, Methods in Practice, Equivalent Syntax) Strings & Lists: What Are Strings? & Declaring Strings in Python ( String and Alphabets, Unicode Characters, Special Characters, Three Ways to Declare Strings)- String Concatenation and Slicing in Python & String Searching in Python ( String Concatenation, String Slicing: Individual Characters, String Slicing: Substrings, Negative Indices, The In Operator, The Find Method, Parameters of the Find Method)- What Are Lists? & Tuples in Python ( Properties of Lists, List Synonyms, Declaring Tuples, Reading Tuples, Usefulness of Tuples, Nesting Tuples )File Input and Output: What Is File Input and Output? & Reading, Writing and Appending ( Output Complements Input, File Types, Getting Started: Opening and Closing Files, Reading, Writing, Appending)- Writing Files in Python & Reading Files in Python ( Simple File Writing, Writing Lists, Another Way to Output, Appending to Files, Simple File Reading, Loading into Lists, Save and Load Functions)-Files ( Preparing to Save and Load, Global Variables, The Save Command, The Load Command).

**UNIT : V** **LECTURE HOURS: 10**

Objects & Algorithms What Are Objects? & Objects and Instances in Python ( What Are Objects?, Objects vs. Instances, Declaring a Class , Creating Instances, Objects vs. Dictionaries )-Encapsulating Methods in Classes & Encapsulating Methods in Python ( Encapsulating Methods, Common Method Types: Constructors and Destructors, Common

Method Types: Getters and Setters, Constructors in Python, Getters and Setters, Encapsulating Other Functions )-Advanced Topics in Classes in Python & Polymorphism and Inheritance and Abstraction, Oh My! ( Combining Classes, Instance Assignments, Instances as Arguments, Making Actual Copies, Abstraction, Polymorphism, Inheritance)- What Are Algorithms? & Complexity and Big O Notation ( Famous Algorithms, Algorithms and Programming Languages, Complexity in Algorithms, Big O Notation, Common Big O Values)- Recursion & Sorting Algorithms ( Simple Recursion: Factorial, Intermediate Recursion: The Fibonacci Series, Advanced Recursion: Directory Exploration, Bubble Sort).

**Text Book:**

T1. Introduction to Computing: Python, McGraw-Hill Education LLC, David Joyner, 1st Edition, 2016.

Course Provider: www.edx.org, Course Name: Introduction-computing-using-python, Instructor Name:David Joyner, University Name: Georgia Tech University, CourseLink: https://www.edx.org/course/introduction-computing-using-python-gtx-cs1301x

**WEB REFERENCE(S):**

LearnPython.org

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

https://interactivepython.org/runestone/static/pythonds/index.html

**Reference Book(S):**

R1. How to Think Like a Computer Scientist: Learning with Python 3 by Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, 3rd edition, 2012

R2. Introduction to Computer Science using Python, Wiley India Publication, Charles Dierbach, 2015.

R3. Beginning Python, Wiley India Publication, James Payne, 2011

**SEMESTER III**
**MAJOR PAPER - IV**
**DATA STRUCTURES**

**Objectives:**

- To enable the students to understand the properties of various data structures.
- Identify the strength and weakness of different data structures
- Design and employ appropriate data structures for solving computing problems

**Pedagogy:**

- The teaching methodology is through lecture, PPT, group discussion, seminars and demonstrations.

| TOTAL HOURS/WEEK – 4 + 1 | | | |
|---|---|---|---|
| TOTAL HOURS/SEMESTER – 75 | | | |
| CIA | CO-CURRICULAR ACTIVITIES | LECTURE | TUTORIAL |
| 10 | 5 | 48 | 12 |

**COURSE OUTCOME**

| No. | Course Outcome | PSOs | Cl. Ses | CL |
|---|---|---|---|---|
| CO1 | Understand the basic concepts of Data Structures | PSO4 | 10 | U |
| CO2 | Understand the basic concepts of Recursion, Backtracking | PSO4 | 10 | U |
| CO3 | Implement Linked list using python | PSO4 | 10 | AP |
| CO4 | Demonstrate the stack ADT using linked list, array data structure and interpret the algebraic expressions using stack | PSO4 | 10 | AP |
| CO5 | Demonstrate the queue ADT using linked list, array data structure | PSO4 | 10 | AP |
| CO6 | Understand the operations using tree ADT and solve the problems using tree | PSO4 | 10 | AP |

**UNIT I** **LECTURE HOURS:10**

Data Structures: Introduction(Variables, Datatypes, Data Structures)- Abstract Data Types (ADTs)( Abstract Data Types)Algorithm: Introduction(What is an Algorithm? Why the Analysis of Algorithms, Goal of the Analysis of Algorithms. How to compare Algorithms)- Analysis(Type of Analysis, Asymptotic Notation, Asymptotic Analysis)Recursion: Introduction(What is a Recursion, Why Recursion, Format of a Recursive function)- Recursion and memory(Visualization)- Recursion Vs Iteration(Example Algorithms of Recursion) Backtracking: Introduction(What is Backtracking? Example Algorithms of

Backtracking)

**UNIT II** **LECTURE HOURS:10**

Linked Lists: Introduction (What is a Linked List?) - Linked List ADTs (Operations and why linked list?)Arrays: Arrays Overview (Why constant time for accessing array elements? Advantages and Disadvantages of Arrays, Dynamic Arrays) - Linked List Overview (Advantages and Disadvantages of Linked Lists)Singly Linked List: Creation and Traversal (Creating and Traversing the Linked list) - Singly Linked list Insertion (Inserting a new node at the beginning, end and middle of the list) - Singly Linked list Deletion (Deleting node at the beginning, end and middle of the list)Variations of Linked List: Circular Linked List (Insertion and deletion of Circular linked list) - Doubly Linked List (Insertion and deletion of Doubly Linked list)

**UNIT III** **LECTURE HOURS:10**

Stack: Introduction(What is a Stack? How stacks are used) - Stack ADT (Operations) Applications: Expressions (Infix, postfix, Prefix) - Evaluating Arithmetic expressions (Conversion of Infix to Postfix, Evaluation of a Postfix expression.)

Problems: Evaluating Arithmetic expressions(Infix to postfix conversion, Postfix Evaluation)

Implementation: Array Implementation (Array Implementation) - Linked List Implementation (Linked List Implementation) - Comparison of Implementations(Comparing Array Implementation and Linked List Implementation)

**UNIT IV** **LECTURE HOURS:10**

Queues: Introduction (What is a Queue? How are queues used) - Queue ADT (Operations) Queue Implementation: Array Implementation (Array Implementation) - Linked List Implementation (Linked List Implementation)Variations of Queue: Circular Queue, Deque (Introduction and working of queue) - Problems on Queue (Insertion, Deletion of queue)

**UNIT V** **LECTURE HOURS:10**

Trees: Introduction (What is a tree?) – Glossary (Glossary)Binary Trees: Types of Binary Tree (Strict, Full and Complete Binary tree ) - Properties of Binary Trees (Properties of

Binary Trees) - Operations on Binary tree (Insert, Delete, Search and traverse)Binary Tree Traversals: Preorder Traversal (Preorder Traversal) - Inorder Traversal (Inorder Traversal) - Postorder Traversal (Postorder Traversal) - Level Order Traversal (Level Order Traversal)Types of Trees: Generic Trees (Representation of generic trees) - Threaded Binary Trees (Structure) - Expression Trees (Expression Trees)

**TEXT BOOK:**

T1.NarasimhaKarumanchi: " Data Structure and Algorithmic Thinking with Python", CareerMonk Publications,2016

(Unit 1 – Chapter 1,2, Unit 2 – Chapter 3, Unit 3 – Chapter 4,5, Unit 4 – Chapter 6, Unit 5 – Chapter 6,7)

T2.G A V Pai : "Data Structures and Algorithms: Concepts, Techniques and Applications" , McGraw Hill Education (India) Pvt Ltd.

NPTEL Course (National Programme on Technology Enhanced Learning) Joined initiative of the IITs and IISC.

"Programming, Data Structures and Algorithms in Python", Dr. MadhavanMukund, Professor and Dean of Studies,

https://onlinecourses.nptel.ac.in/noc16_cs11

## SOFTWARE ENGINEERING

**OBJECTIVIES:**

- Develop in students a critical understanding of issues related to the engineering of large complex software systems.
- Know how to develop a set of requirements,
- Apply rigorous software analysis.
- To design, code & test their work.

**PEDAGOGY:**

- Teaching aids used are Black board, OHP, Projector, Demonstration and Group Discussion.

| TOTAL HOURS/WEEK - 5 | | | |
|---|---|---|---|
| TOTAL HOURS/SEMESTER - 75 | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** |
| **10** | **5** | **60** | **-** |

**UNIT I (LECTURE HOURS: 13)**

**The Evolving Role Of Software And Software**

**The Evolving Role Of Software** - Explanation

**Software: Software Characteristics and Applications** - Definition, Explanation about Software Characteristics.

**Software Crisis And Myths**

**Software Crisis** - Explanation about crisis.

**Software Myths** - Definition, Explanation about Management myths, Customer Myths, Practitioner's Myths.

**Software Engineering: A Layered Technology**

**Process, Methods, and Tools , A Generic View of Software Engineering** - Explanation about Process, Methods andTools, A Generic View of Software Engineering

**The Software Process**

**The Software Process** - Definition, Explanation about Process Maturity, diagram- A Common Process Framework.

**Software Process Models And The Prototyping Model**

**Software Process Models** - Definition, Explanation about Process Model, Diagram - The phases of a problem, The Phases within Phases.

**The Prototyping Model** - The Linear Sequential Model - Explanation, Diagram

**The Rad Model And Evolutionary Software Process Models**

**The RAD Model** - The Prototyping Model, The RAD Model-Explanation, Diagram.

**The Incremental Model and Spiral Model** - The Incremental Model, The Spiral Model - Explanation, Diagram.

## UNIT II (LECTURE HOURS: 13)

**Requirements Engineering**

**Requirements Elicitation, Requirements Analysis and Negotiation, Specifications, System Modeling, Validation, Management.** - Explanation about Requirements Elicitation, Requirements Analysis and Negotiation, Specifications, System Modeling, Validation, Management.

**System Modeling**

**System Modeling** - Explanation, Diagram - Sample Model Template, System context Diagram, System Flow Diagram.

**Requirements Analysis And Requirements Elicitation For Software**

**Requirements Analysis** - Explanation, Diagram - Analysis as a bridge between system engineering and softwaredesign.

**Initiating the process, Facilitated Application Specification Techniques, Quality Function Deployment, Use-cases**

- Explanation about Initiating the process, Facilitated Application Specification Techniques, Quality Function Deployment, Use-cases.

**Software Prototyping And Specification**

**Selecting the Prototyping Approach, Prototyping Methods and tools.** - Explanation about Selecting the Prototyping Approach, Prototyping Methods and tools.

**Specification Principles, Representation, The Software Requirements Specification** - Explanation about Specification Principles, Representation, The Software Requirements Specification.

**The Elements Of Analysis Model And Data Modeling**

**The Elements of Analysis Model** - Explanation about DFD, STD, Diagram - The Structure of the analysis Model.

**Data Modeling** - Explanation about Data Objects, Attributes, and Relationships, Cardinality and Modality, Entity /Relationship Diagrams.

**Functional Modeling And Information Flow**

**Functional Modeling And Information Flow** - Explanation about Data Flow Diagrams, Extensions for Real- Time Systems, Ward and Mellor Extensions, Hatley and PribhaiExtensions.

**The Mechanics Of Structured Analysis And The Data Dictionary**

**The Mechanics of Structured Analysis** - Explanation about Creating an Entity / Relationship Diagram, Data Flow Model, Control Flow Model, Control Specification, Process Specification.
**The Data Dictionary** - Definition, Explanation about Data Dictionary.

**UNIT III (LECTURE HOURS: 11)**

**Design For Object Oriented System And The System Design Process**
**Conventional Vs OO Approaches, Design Issues, The OOD Landscape, A Unified Approach to OOD** - Explanation Diagram- The OOD design pyramid, Conventional Vs OO Approaches, Diagram - Translating OOA model into an OOD model, Design Issues, The OOD Landscape, A Unified Approach to OOD, Diagram - Process flow for OOD. **Partitioning the Analysis Model, Concurrency and Subsystem Allocation, The Task Management Component,**
**User Interface Component, The Data Management Component, The Resource Management Component, Inter subsystem Communication** - Explanation, Partitioning the Analysis Model, Concurrency and Subsystem Allocation, The Task Management Component, User Interface Component, The Data Management Component, The Resource Management Component, Inter subsystem Communication.

**Software Design & Engineering And The Design Process & Principles**
**Software Design and Software Engineering** - Explanation, Diagram-Translating the analysis model into a software design
**The design process** - Explanation, Explanation about Design and Software Quality, The Evalution of Software Design.

**Design Principles** - Explanation about Principles for software Design.
**Design concepts**
**Design Concepts** - Explanation, Explanation about Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure, Software Procedure, Information Hiding.

**Effective modular design and design heuristics for effective modularity**
**Effective Modular Design** - Explanation, Explanation about Functional Independence, Cohesion, Coupling.
**Design Heuristics for Effective Modularity** - Explanation about set of Heuristics.

**UNIT IV (LECTURE HOURS: 12)**

**Mapping requirements into a software architecture and Transform & transaction mapping**
**Transform flow, Transaction flow** - Explanation, Explanation about Transform flow, Transaction flow.

**Transform Mapping -An Example, Design Steps** - Explanation, Explanation about An Example, Design Steps. **Transaction Mapping - An Example, Design Steps** - Explanation, Explanation about An Example, Design Steps. **User Interface Design And Activities**

**Interface Design Models, The User Interface Design Process** - Explanation about Interface Design Models, The User Interface Design Process.

**Defining Interface Objects and Actions, Design Issues.** - Explanation about Defining Interface Objects and Actions, Design Issues.

**Software Testing Fundamentals, White Box Testing, Basis Path Testing, Control Structure Testing, And Black Box Testing**

**Software Testing Fundamentals - Testing Objectives, Principles, Testability** - Explanation about Testing Fundamentals, Testing Objectives, Testing Principles, Testability

**White Box Testing** - Explanation about White box testing, The Nature of Software Defects.

**Basis Path Testing - Flow Graph Notation, Cyclomatic Complexity, Deriving Test Cases, Graph Matrices** - Explanation about Basic Path Testing, Flow Graph Notation, Cyclomatic Complexity, Deriving Test Cases, Graph Matrices, Flowchart.

**Control Structure Testing - Condition Testing, Data Flow Testing, Loop Testing** - Explanation, Explanation about Condition Testing, Data Flow Testing, Loop Testing.

**Black Box Testing - Graph-Based Testing Methods, Equivalence Partitioning, Boundary Value Analysis, Comparison Testing, Orthogonal Array Testing** - Explanation about Black Box Testing, Graph-Based Testing Methods, Equivalence Partitioning, Boundary Value Analysis, Comparison Testing, Orthogonal Array Testing.

## UNIT V (LECTURE HOURS: 11)

**A Strategic Approach To Software Testing, Validation Testing, System Testing, And The Art Of Debugging**

**Strategic Approach to Software Testing - Verification and Validation, Organizing for Software Testing, A Software Testing Strategy, Criteria for Completion of Testing** - Explanation about Strategic Characteristics, Verification and Validation, Organizing for Software Testing, A Software Testing Strategy, Criteria for Completion of Testing

**Validation Testing - Validation test Criteria, Configuration Review, Alpha and Beta testing.** - Explanation about Validation Testing, Validation test Criteria, Configuration Review, Alpha and Beta testing.

**System Testing - Recovery Testing, Security Testing, Stress Testing, Performance Testing** - Explanation about System Testing, Recovery Testing, Security Testing, Stress Testing, Performance Testing.

**The Art of Debugging - The Debugging Process, Psychological Considerations, Debugging Approaches.** - Explanation about

Debugging, The Debugging Process, Psychological Considerations, Debugging Approaches. **Software Quality**

**McCall's Quality Factors, FURPS, ISO 9126 Quality Factors, The Transition to a Quantitative view** - Define Quality, Explanation about McCall's Quality Factors, FURPS, ISO 9126 Quality Factors, The Transition to a Quantitative view. **The Software Reengineering And Reverse Engineering**

**Software Reengineering -Software Maintenance, A Software Reengineering Process Model** - Explanation about Software Maintenance, A Software Reengineering Process Model, and Diagram.

**Reverse Engineering - Reverse Engineering to Understand Processing and Data, Reverse Engineering User Interfaces.** - Explanation about Reverse Engineering Process and Diagram, Reverse Engineering to Understand Processing, Reverse Engineering to Understand Data, Reverse Engineering User Interfaces.

**Building blocks for case and a taxonomy of case tools**

**Building Blocks for CASE** - Explanation about CASE Building Blocks and Diagram, Diagram about Integration Options.

**A Taxonomy of CASE Tools** - Concept, Function Classification about CASE Tools,

**Text Books :**

Software Engineering | Edition:5th Edition | TMH Publishers | Roger S Pressman(2001)

**Reference Books :**

A discipline for Software Engineering | Edition: | Pearson Education Publishers | Watts S Humphrey(2001) Software Engineering | Edition:6th Edition | Pearson Education Publication | Ian Somerville(2001)

## SEMESTER IV

| | |
|---|---|
| Course Title :**DESIGN AND ANALYSIS OF ALGORITHMS (T)** | Course Code : **43A** |
| Semester : **IV** | Course Group : **DSC-** |
| Teaching Scheme in Hrs (L:T:P) : **4:0:0** | Credits : **4** |
| Map Code :**E (Theory-Programming)** | Total Contact Hours : **60** |
| CIA : **25 Marks** | SEE# : **75** |
| Programme : **BSc-CS/BCA/B.Sc(IT)** | # - **Semester End Exam** |

**COURSE OUTCOME**

| No. | Course Outcome | PSOs | Cl. Ses | CL |
|---|---|---|---|---|
| CO1 | Understand the time complexity of various data structures | PSO4 | 12 | U |
| CO2 | Analyze the time complexity using Array and List | PSO4 | 6 | AP |
| CO3 | Describe and Analyze Heap Data Structure | PSO4 | 6 | AP |
| CO4 | Understand and Demonstrate the Searching and Sorting techniques | PSO4 | 12 | AP |
| CO5 | Describe and Demonstrate Binary Search Tree and Graphs | PSO4 | 12 | AP |
| CO6 | Understand Spanning Trees, Greedy Algorithms, Weighted Graphs and analyze the time complexity | PSO4 | 12 | AP |

**UNIT I** **LECTURE HOURS:12**

Algorithm: Introduction (What is Algorithm?,How does it look like?, Example, Why Algorithms?) - Design of Algorithm (What is design of an algorithm?, Iterative and Recursive algorithms, The design steps, The design techniques, The role of Data structures in algorithm design.)Asymptotic Complexity: Analysis of Algorithms (What is Analysis?, Why Analysis?)- How to do Analysis? Efficiency and Running time an Algorithm (Definition, Big 'O' notation Measuring the running time of an algorithm,

How to find an efficiency of an algorithm?)-Complexity (Definition, Space and Time complexity, Difference between performance and complexity, Complexity classes)-Worst case of an algorithm (What is worst case of an algorithm? How to determine complexity?)- Calculating complexity with iterative version (statements, If statements, single, double and triple loop (for) example)- Calculating complexity with Recursive version(While loop, Towers of Hanoi puzzle)

The two basic data structure: Arrays and Lists(Introduction, storage of values, what is Array and List in storage point of view?)- Array(Access an element with offset, Insert an element, the worst case input,Delete an element)- List(Access an element, Inserting an element using plumbing, Deleting an element)-Array vs. List(Difference between Array and List in complexity point of view)

Analysis of Stack and Queue: Implementing stack using Array and List(Create, Inserting an element, Deleting an element, Traversing the datastructure)- Implementing queue using Array and List(Create, Inserting an element, Deleting an element, Traversing the datastructure)

**UNIT II** **LECTURE HOURS:12**

Hash Table: Hash Table(Introduction, Storing values in Hash Table, Hashing)- Hash function(Choosing a good Hash function, Perfect Hash function, Load Factor)- Basic Operations and their Time complexity(Search, Insert, Delete)-Collision Resolution techniques(Linear Probing ,Separate Chaining, Applications of Hash)-Applications(Hash Applications)

Heap: Heap Structure(Introduction, Heapify and it's analysis)- Types of heap(Max Heap, Min Heap)- Max Heap and it's Analysis(Find ,max, insert, delete)- Min Heap and it's Analysis(Find, min, insert, delete)- Other Heap operations(Merge, Meld, shift up and shift down)

**UNIT III** **LECTURE HOURS:12**

Searching: Search Problem(Definition)- The Unsorted case(Linear Search – example)-The sorted case(Binary Search – example)- Comparison(Comparison)

Sorting: Introduction(Definition, The need for sorting, Advantages)- Selection sort(sorting using swap)- Analysis(Recursive and Iterative versions)- Insertion Sort(The swap example)- Analysis(Recursive and Iterative versions)- Comparison of sequential sorting(Selection sort Vs. Insertion Sort Vs. Bubble Sort)

Divide and Conquer: Introduction(The Divide and Conquer Strategy, Example, Advantages, Disadvantages)- Merge Sort(Example)- Analysis(Merge, Mergesort, Shortcomings)- Quick Sort(Introduction, Sort with Pivot, Tony Hoar's actual partitioning method)- Analysis(Randomization, Applications)

Stable Vs. Unstable sort: Stable Sort(Definition, Example)- Unstable Sort(Definition, Example)- Which sorting is best?( Comparison)

**UNIT IV** **LECTURE HOURS:12**

Binary Search Tree: Operations(The structure of a BSTree, Find(v), Minimum, Maximum, Predecessor, Successor, insert, delete)- Analysis(Analysis of all operations)

Graphs: Formal Representation(Directed Edge, Undirected Edge)- Working with Graphs Representation(Adjacency Matrix, Adjacency List)

Graph Explorations: Breadth First Search(BFS)( Example, Algorithm)- Analysis(Adjacency Matrix, Adjacency List)- Depth First Search (DFS)( Example, Algorithm)- Analysis(Adjacency Matrix, Adjacency List)- Applications of BFS and DFS(Properties of DFS, Identifying connected components, Identifying cycles)

**UNIT V** **LECTURE HOURS:12**

DAG (Directed Acyclic Graph): Introduction(Directed Cycles, DAG definition, Features of DAG, Example)Greedy Algorithm: Greedy Strategy(Introduction, Comparison to the other techniques, Advantages)Shortest Path: Shortest Path(Weighted Graph, classification) - Single source shortest path (Dijksta's Algorithm, Example, Analysis)Spanning Trees: Tree, Spanning Tree (Definition with example) - Spanning Tree with costs (Example)Minimum Cost Spanning Tree: Minimum Cost Spanning Trees (Prim's Algorithm)- Minimum Cost Spanning Trees(Kruskal's Algorithm)- Minimum Cost Spanning Trees(Analysis)

**REFERENCES:**

R1.Grokking Algorithms - An Illustrated guide for programmers and other curious people by Aditya Y. Bhargava, Manning Publications, 2016

R2.Data Structures and Algorithms made easy by NarasimhaKarumanchi, Career Monk Publications, 2014

R3.Computer Algorithms by Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, Computer Science Press, 2011

NPTEL: http://nptel.ac.in/courses/106106131

https://www.coursera.org/courses?languages=en&query=data+structures+and+algorithms

## SEMESTER – IV

## ELECTIVE -I

## COMPUTER NETWORKS

**OBJECTIVES:**

- Understanding the Basic Concepts of Networks
- Understanding the different Types of Cables, Connectors, and Networking Devices.
- Getting familiarization with Ethernet and IP Address Configurations.

**PEDAGOGY:**

- Teaching aids used are Black board, OHP, Projector, Demonstration and Group Discussion.

| TOTAL HOURS/WEEK-5 | | | |
|---|---|---|---|
| TOTAL HOURS/SEMESTER-75 | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** |
| **10** | **5** | **60** | **-** |

**UNIT I (LECTURE HOURS: 12)**

**Networking Overview**

**Introduction to Networking, Network Types** - LAN,WAN,MAN

**Networking Terms, Networking Facts,Practice Questions** - Transmission Media, Network interfaces, Protocols

**Networking Topologies**

**Network Topologies,Topology Facts , Practice Questions** - Bus, Ring, Star, Mesh, Terminators

**The OSI Model**

**The OSI Model,The OSI Model Facts** - Provides a common language and reference points for network professionals,Aids in troubleshooting, Provides modularity.

**The OSI Model Layers,The OSI Model Communications** - HTTP, Telnet, FTP, TFTP, SNMP, Formats or "Presents"the data

**OSI Layers Facts,Practice Questions** - Application, Presentation, Session, Transport, Network, Data Link, Physical.

**Network Signaling**

**Network Signalling,Transmission Systems** - What is the purpose of Network signalling, Difference between basebandand broadband.

**Network Signalling Facts,Practice Questions** - Return-to-Zero, Non-Return-to-Zero, Manchester, De-Multiplexing.

**Network Protocols**

**TCP/IP Protocol Suite,Common Network Services** - How does TCP differ from UDP, How does a protocol suite differfrom a protocol.

**Common TCP/IP Protocols,Practice Questions** - Web Services, HTTP, HTTP over SSL, SSL, Security Protocols, TLS,File Transfer, POP3, IMAP, IMAP4

**Numbering Systems**

**Numbering System,Numbering System Facts,Practice Questions** - What are the possible values in a binary number,Difference between a binary number and a hexadecimal numbering system , Binary (Base 2), Octal (Base 8), Hexadecimal (Base 16)

**UNIT II (LECTURE HOURS: 12)**

**Twisted Pair**

**Twisted Pair,Twisted Pair Facts, Practice Questions** - Why are wires twisted together in twisted pair cables,Difference in CAT3 and CAT5 cables, CAT 3, CAT 5E, CAT 6, CAT 6A.

**Coaxial**

**Coaxial,Coaxial Cable Facts,Practice Questions** - What is the function of the wire mesh in coaxial cables, which partof the cable is used to carry data, RG58, RG59, RG6, BNC

**Fiber Optic**

**Fiber Optic,Fiber Optic Facts, Practice questions** - Totally immune to EMI, Very expensive, Highly resistant toeavesdropping, Difficult to work with.

**Wiring Implementation**

**Twisted pair cable construction,Cable construction facts** - Straight-through, Crossover, What is a patch panel used for.

**Wiring Distribution,Using Punchdown blocks** - Using Punchdown

**Wiring Distribution Facts, Practice Questions** - Demarcation point, Main Distribution Frame, Demarc extension, Horizontal cross connect, vertical cross connect

**Troubleshooting Network Media**

**Troubleshooting Copper Wires Issues, Copper Wiring Troubleshooting facts** - EMI and RFI, Crosstalk, Near End Crosstalk, Far end Crosstalk.

**Troubleshooting Fiber opting Wiring Issues, Fiber Optic Wiring Troubleshooting facts** - Connectors, Polishing, Cabling, Media adapters, Attenuations

**Troubleshooting tools, Troubleshooting tool facts, Practice Questions** - Loopback plug, Smart jack, cable tester, time-domain reflectometer

**UNIT III (LECTURE HOURS: 12)**

**Network Adapters**

**Network Adapters** - Connect a media convertors, What does FF-FF-FF-FF-FF-FF indicates **Network Adapter Facts** - A transceiver, Demodulation, Transceiver modular, Media convertor**Connect a media converter** - Port 23 is Tx, Port 24 is Rx.

**Practice Questions** - Practice questions.

**Network Devices**

**Network Devices** - At which OSI model layer do wireless access points operate, what type of device do you use to translate from one network architecture to another

**Network Connection Device Facts** - Hub, Bridge, Switch, Wireless AP

**Practice Questions** - Practice Questions

**Internetwork Devices**

**Internetwork Devices, Internetwork Devices Facts** - Router, Firewall, Layer 3 switch

**Practice Questions** - Practice Questions

**UNIT IV (LECTURE HOURS: 12)**

**Ethernet**

**Ethernet Architecture** - What logical topologies are supported on an Ethernet network, what device is used to enable fullduplex, what is the purpose of the backoff on Ethernet network.

**Ethernet Facts** - Topology, Networking Devices, Transmission Media, Media Access Method.

**Practice Questions** - Practice Questions

**Ethernet Specifications**

**Ethernet Specifications, Practice Questions** - Reconnect to an Ethernet Network, Ethernet, 10BaseT, 10BaseFL,100BaseTX.

**Connecting Network Devices**

**Connecting Devices, Device Connection Facts** - Domain 3.0 Ethernet, When would you use a rollover cable?

**Practice Questions** - Practice questions

**Troubleshooting Physical Connectivity**

**Troubleshooting the Fault Domain** - Explore physical connectivity.

**Fault Domain Trobleshooting Facts,Troubleshooting the Link Status** - Bus, Star, Ring, Mesh

**Link Status Troubleshooting facts,Exploring Physical Connectivity** - Unlit, Activity, Collision, Red/Amber, Solid Green.

**Practice Questions** - Practice Questions

**UNIT V (LECTURE HOURS: 12)**

**IP Addressing**

**IP Addressing,IP Address Facts** - What is an Octet, Configure IP address, How does VLSM works, IP address is a32-bit binary numbers.

**Subnetting ,Subnetting Facts** - Increase the number of devices, Decrease the number of devices, subnetting usescustom subnet masks.

**Variable Length Subnetting facts** - Class A, Class B, Class C, classless addresses, Partial Subnetting, VLSM.

**IP Address Assignment** - Subnet, Host address, Default Gateway, TCP/IP settings

**IP Addressing Facts** - DHCP, Static (manual) assignment.

**Practice Questions** - Practice Questions

**Alternate IP Addressing**

**APIPA -** Configure an alternate IP configuration.

**Alternate IP Addressing facts** - APIPA, Alternate IP Configuration.

**Practice questions.** - Practice questions.

**Text Books :**

CCENT/CCNA ICND1 100-105, Official Cert Guide Wendell Odom, CCIE NO.1624 | Edition:ICND1 100-105 | Ciscopress.com | WendellOdom(2016)

**Reference Books :**

Computer Networks | Edition:5th | Prentice Hall | Andrew S. Tanenbaum AND David J.Wetherall(2014)

# SEMESTER-V
# SBC- II
# AngularJS

**OBJECTIVES:**

- Understand the design of single-page applications and how AngularJS facilitates their development.
- Properly separate the model, view and controller layers of application and implement using AngularJS.
- Angular Material is both a UI Component framework and a reference implementation of Google's Material Design Specification.
- It provides a set of reusable, well-tested, and accessible UI components based on Material Design.

**PEDAGOGY:**

Teaching aids used are Black board, OHP, Projector, Demonstration and Group Discussion.

| TOTAL HOURS/WEEK - 4 | | | | |
|---|---|---|---|---|
| TOTAL HOURS/SEMESTER - 60 | | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** | **TOTAL** |
| **8** | **4** | **48** | **-** | **48** |

## UNIT I (LECTURE HOURS: 9)

**Introduction & Architectural concepts**

**Model-View-Controller (MVC)** - MVC,MVW, Language, Platform, and Purpose of the application

**DIRECTIVE AND USING ANGULARJS BUILT-IN DIRECTIVES**

**The ngApp directive** - The root of an AngularJS application to bootstrap the framework

**The ngController directive** - The view and controller start to share the same scope and are ready to work together

**The ngBind directive** - Span element and replaces the content of the element with the results of the provided expression

**The ngRepeat directive** - To iterate over arrays and objects. The rows of a table, the elements of a list, and the options of select

**The ngModel directive** - Attaches the element to a property in the scope, binding the view to the model

**The ngClick directive and other event directives** - To bind any custom behavior to the click event of the element

**CREATING OUR OWN DIRECTIVES**

**Template** - The number of times the same snippet of the HTML code repeated over application code

**Template Url** - The snippet to an isolated file and bind it using the templateUrl property

**Replace** - To discard the original element, replacing it by the directive's template

**Restrict** - The directives are restricted to be applied as an attribute to a determined Element

**EXPRESSIONS-BASIC USAGE WITH EXPRESSIONS**

**Date** - A date Value comes from the database or any other source in a raw and generic format

**Filter** - Acting Over an array and applying any filtering criteria

**Number** - To format a string as a number

**Orderby** - Order any array based on a predicate expression. String, array, function

**UNIT II (LECTURE HOURS: 9)**

**FORM VALIDATION**

**Basic validation** - The ngRequired directive, to intimate the validation process that the field is actually required

**Understanding the $pristine and $dirty properties** - Pristine means purity that the field wasn't touched by anyone. After it's been touched for the first time, it becomes dirty

**The $error object** - It accumulates the detailed list of everything that happens with the form

**Dependency injection**

**The level of Dependency** - The object-oriented world, is known as coupling, and indicates the level of Dependencybetween the components

**Creating services**

**Creating services with the factory** - To register the service in the application module that passes two parameters : thename of the service And the factory function

**Creating services with the service** - It uses a constructor function, which is equivalent to using the new operator

**Creating services with the provider** - The provider relies

on the $get function to expose its behavior

**Communicating with the backend**

**HTTP, REST, and JSON** - To interact with the backend was through http with the help of the get and post methods

**Creating a single-page applicationInstalling the module** - The $route service by mapping urls against controllers and views, and parameter passing

**Configuring the routes** - Controller, templateurl, resolve,redirectto

**Rendering the content of each view** - The $route service and is responsible for rendering each template according to the routing mechanism

**Passing parameters** - To inject the $routeparams service, which will provide us with the parameters passed through the url

**Global Behavior** - It can be injected inside any component such as controllers, directives, filters, and services

**$Broadcast** - To communicate between components by the means of a scope

## UNIT III (LECTURE HOURS: 10)

**Introduction**

**What is angularjs material?&Goals & Principles** - UI component framework-Google's material design specification

**AngularJS Material Environmental setup**

**Installing the angularjs material libraries** - Configuring the library files for dependencies **Build a material application (blank shell)** - Configure the folders and files for application **Introduction Material Design Layouts**

**Layout and containers** - Create modern, responsive layouts on top of CSS3 flexbox

**Layout and responsive break points** - Associate breakpoints with mediaQuery definitions using breakpoint alias(es): Layout- xs,gt,sm,gt-sm,etc.

**Layout API and Breakpoint overrides methods** - Simple Layout markup convention. The alias is used as suffixextensions to the Layout API keyword

## UNIT IV (LECTURE HOURS: 9)

**Flex Directive**

**Responsive flex directives** - The flex directive value is restricted to multiples of five, 33 or 66**Additional flex values** - There are additional flex values provide to improve flexibility and to make the API easier tounderstand

**Ordering HTML elements** - Its order position within the layout container. Any integer value from -20 to 20 is accepted

**FLEX API and Breakpoint overrides methods** - Device width when breakpoint overrides default **Add offsets to the preceding HTML Elements** - Flex-offset the margin-left offset is applied **Child**

**Alignment**

**Set Child alignments within the layout container** - The children aligned in the layout's direction and perpendicular to the layout's direction
**Layout-Margin** - Adds margin around each flex child
**Layout-Padding** - Adds padding inside each flex child
**Layout-wrap** - A non-trivial group of flex elements using layout-wrap

**Layout-fill** - Forces the layout element to fill its parent container

**Show & Hide** - The show hide APIs to responsively show or hide elements

**UNIT V (LECTURE HOURS: 11) DIRECTIVES**

**Autocomplete** - To search for matches from local or remote data sources

**Bottom Sheet** - Display content by click one of the buttons below
**Button** - Button directive with optional ink ripples

**Card** - Directive is a container element used within

**Checkbox** - Directive is used like the normalangular checkbox

**Chips** - Component used within and is responsible for rendering individual chips

**Content** - Directive is a container element useful for scrollable content

**Dialog** - The dialog's template must be inside this element

**Datepicker** - The dialog's template must be inside this element

**Divider** - Dividers group and separate content within lists and page layouts using strong visual and spatial distinctions

**Input** - Element as a child of an It allows to build complex forms for data entry

**Icon** - Directive makes it easier to use vector-based icons **Input-Container** - The parent of any input or textarea element **List** - Directive is a list container for 1..n tags

**Nav Bar** - Directive renders a list of material tabs that can be used for top-level page navigation

**Radio Button** - Directive is the child directive required to be used within elements

**Sidenav** - Component that can be opened and closed programatically

**Select** - Component can be used within a or as a stand alone component by using the md-no-underline class

**Subheader** - Directive creates a sticky subheader for a section

**Toolbar** - Place a toolbar in application

**Tabs** - Tab with a label and optional view content **Whiteframe** - Apply an elevation shadow to an element **User-card** - Card avatar,Class for user image

**Text Books :**

Angular JS Essential | Edition: | Packt Publication | Rodrigo Branas( )

**Website:**
https://material.angularjs.org
https://www.codeschool.com

## ANDROID PROGRAMMING

| No. | Course Outcome | POs & PSOs | Cl. Ses | CL |
|---|---|---|---|---|
| CO1 | Extend work with Fragments & Concepts | PO1, PSO1,PSO2 ,PSO3 | 18 | AP |
| CO2 | Extend work with Threads & Files | PO1, PSO1,PSO2 PSO3 | 8 | AP |
| CO3 | illustrations about Files, Adapters and intents. | PO1, PSO1 ,PSO2 PSO3 | 10 | AP |
| CO4 | Perform techniques to work with SQLite databases & Classes | PO1, PSO1 ,PSO2 PSO3 | 18 | AP |
| CO5 | Extend work with tabs & customer adapters | PO1, PSO1 ,PSO2 PSO3 | 18 | AP |
| CO6 | Creation and working with deploy an App | PO1, PSO1 ,PSO2 PSO3 | 18 | AP |

**UNIT – I** **LECTURE HOURS: 18**

An introduction to fragments : Single -pane and Multi-pane Layouts (Fragment to define part of the user interface for an activity) - How to use support libraries (The support library for API 4 Makes the fragment class compatible with all modern versions) - The life cycle method of fragments (Create view method, Destroy view method ) - How to use single pane layouts for small screens : How to create the layout for a fragment (Start xml code for a fragment, use Graphical layout editor for a fragment) - How to create the class for a fragment (On Create method and On Create View Method) - How to display a fragment in an activity (Add fragment element and use name attribute to specify fully qualified name for the class.) - How to create a preference fragment. (Fragment needs to extend Preference Fragment class) - How to display a preference fragment in an activity. (Use add Preference Form Resource method to add the Preferences defined in XML file.) - How to use Multi-pane layouts for large screens : How to add multiple fragments to a layout. (Add two or more fragment elements to a layout file ) - How to detect large screens (Create a values directory) - How to detect screen width (Create a values directory that uses smallest-width qualifier.) - How to control the soft keyboard (Delete request Focus element from body of Edit Text element) - Other skills for working with fragments : How to get a reference to a fragment (Fragment manager object) - How to replace one fragment with another (Fragment manager object).

**UNIT – II** **LECTURE HOURS: 18**

An introduction to the news reader app : The user interface (The items activity displays the title of the news feed followed by a list of items) - The XML for an RSS Feed (RSS (Rich site Summary),Standardized XML) - How to work with Threads : How threads work (Single sequential flow of control within a program) - How to execute asynchronous tasks (This task runs in a separate thread in background and does not tobe synchronized.) - How to execute timed tasks (Use TimerTask and Timer classes to create a thread that execute at specific interval) - How to update UI thread (Use post method of a view object to execute any runnable object on UI thread) - How to work with files : How to download a file from the internet (Use Java and Android APIs to download a file by reading input from Internet and writing output to file system.) - How to parse an XML file (Use SAX to parse XML files.) - The RSS Feed handler class (Extends Defaulter Handler which contains override ,First two statements define instance variables) - The RSS Feed class (Stores data for RSS feed which includes feed's title and publication date) - The RSS Item class (Simple data format objects to change the format for the date.) - How to work with Adapters : How to create the Layout for a list view (Begins by using TextView widget to display the title of news feed, uses ListView Widget to display items) - How to use an adapter to display data in a list view (Use SimpleAdapter class to display data.) - How to handle events for an adapter. (onItemClick method ) - How to work with Intents : How to pass data between activities. (Explicit intent specify a component ) - How to view a URL in a web browser. (Implicit intent ,code passes ACTION_VIEW constant as first argument ) - How to dial or call a phone number. (ACTION_DIAL or ACTION_CALL constant ) - The news reader app : The activity _items layout (Listview Widget, listview layout) - The items activity class (The RSSfeed Object, PostExecutive Method) - The file IO Class (Content Object, Content object parameter) - The activity_ item layout (Textview Widget) - The item activity class (The item Activity class, Uri Object).

**UNIT – III** **LECTURE HOURS: 18**

An introduction to databases : User interface for TaskList app (App provides personal and business,to display click on tabs, edit a task.click Delete tasks button to remove all completed tasks.) - An introduction to SQlite. (Three data types (String,lone,double)) - An introduction to task list database. (Explanation - CREATE TABLE, DROP TABLE) - The business

objects for the task list app (The list class defines an object that can store data from the list table ) - How to create a database class : How to define the constants for a database. (Define constants for name and version. Tables names, column names, column numbers) - How to define the sql statements that create a database (The TaskDB class) - How to create or upgrade a database. (Build- CREATE TABLE DROP TABLE, ALTER TABLE.) - How to open or close a database connection. (Use methods – getReadableDatabase (), get Write Database (), close ()) - How to add public methods to database class : How to retrieve multiple rows from a table. (Use WHERE clause) - How to retrieve a single row from a table. (Works by getTask Method) - How to get data from a cursor. (Use getXxx method to get data) - How to insert, update and delete rows (Use methods – Insert(table,columns,values),update(table,values,where,whereargs),Delete(table,where,where args)) - How to test the database class and clear its data : How to test the database class. (Write code to create an instance of DB class and use public methods. After building display in TextView widget) - How to clear test data from a device. (Use settings app to delete the database for an app.) - How to use DDMS perspective to work with database files. (Select directory and click "push a file".) - How to use the SQLite Database browser. (Use – SQL DB browser and install SQLite DB browser. Create a new DB,open,viewand execute SQL statements).

**UNIT – IV** **LECTURE HOURS: 18**

How to use tabs : How to add the TabManager class to your project (Get TabManager jar file ) - The Layout for an activity that display tabs (TabHost widget defines a container for tabs) - The class for an activity that displays tabs (TabManager object to add one or more tabs) - The class for a fragment that displays tab content (Use Tabhost class to get tag for current tab) - How to use custom adapters : A layout for a list view item (ListView widget used to display list of tasks) - A class that extends the layout for a list view item. (Listview Widget contains complex widgets that extends layout) - A class for a customer adapter. (Extends BaseAdapter class) - A class for a fragment that uses a custom adapter. (Use custom adapter to display appropriate data) - The Task List app : The user interface. (Two list :Personal& business) - The activity_task_list menu. (Add Task item, Delete item) - The TaskListActivity class (onoptionsItemSelected method) - The activity_add_edit and spinner_list layout. (Use spinner to select name ,user can view editable text to edit name and notes.) –

The activity_add_edit menu (Defines menus ,display two items 1.save item 2.cancel item) – Addedit - Activity class (Creates TaskListDB).

**UNIT – V** **LECTURE HOURS: 18**

An introduction to distribution and monetization: How distribution works. (Application marketplace or distribute directly to users via email or website.) - How monetization works. (Getting paid for an app,app billing provides in-app products.) - How to create a release build : How to prepare an app for release. (After testing and debugging,release y final check up) - How to create the signed APK file. (App must store in APK file signed with digital certificate) - How to distribute via a web site. (Create mobile friendly web page) - How to distribute directly to the user : How distribute via email (Attach APK file.) - How to publish on Google play: How to set up a publisher account (Upload APK file and set up publisher account) - How to use the developer console to publish an app. (Name of app,brief description of app,two screenshots of an app,an image of an app.) –

How to view the listing for an app. (Google play and search for an app) - The runmapactivity class (FragmentActivity) - The runtrackerservice class (Instance variables, request location).

**TEXT BOOK:**

T1. Training and Reference Murach's Android programming by Joel Murach , Mike Murach &Associates ,Inc. -Chapters 9,10,13,14,17

**REFERENCE BOOK**:

R1. Beginning Android Application Development by Wei-Meng Lee and Wiley Publishing, Inc. 2011

R2. Android Programming Tutorials by Mark L. Murphy and CommonsWare, LLC., 2011

R3. Android Apps for Absolute Beginners Edition:2 A Press Wallace Jackson(2013)

# SEMESTER-VI.
## ELECTIVE-III
## MONGODB

**Objectives**

To enable the learners to

- Understand the purpose of NOSQL database
- Familiarize the concepts of MongoDB

**Pedagogy**

- Using lecture, group discussion, LCD, Demonstration and seminars

| TOTAL HOURS/WEEK-4+1 | | | | |
|---|---|---|---|---|
| TOTAL HOURS/SEMESTER – 75 | | | | |
| **CIA** | **CO-CURRICULAR ACTIVITIES** | **LECTURE** | **TUTORIAL** | **TOTAL** |
| **10** | **5** | **48** | **12** | **60** |

**UNIT I (LECTURE HOURS: 9) INTRODUCTION TO MONGODB**

**What is MongoDB** - Database, collection, Document

**Document oriented database** - Document

**Features of MongoDB** - Ad hoc queries, indexing, replication, duplication of data, load balancing, map reduce, schema less, javascript, high performance

**NoSql Databases** - NoSql Database, Advantages

**Common terms in MongoDB** - _id, collection, cursor, Database, Document, Field, JSON

**Difference between MongoDB & RDBMS, Sql & Nosql** - Table, row, column, joins, Sql, Nosql

**MongoDB Datatypes** - String, integer, Boolean, Double, Min/Max keys, Arrays, object, Null, Symbol, Date

**Databases and collections**

**views** - Create view, view creation, shared view, drop a view, modify a view, supported operation **Capped collections** - Overview, behavior, restrictions and recommendations, procedures **DOCUMENTS**

**structure** - Field name, field value limit

**Dot notation** - Arrays, embedded documents

**Document limitation** - Document size limit, field order, _id field

**Other uses of the document structure** - Query filter documents, update specification documents, index specification documents

## UNIT I TUTORIAL HOUR (LECTURE HOURS: 3)

**Use command and collection**

**Create a Student Database using "use" Command** - Use command **Create an Employee Database using "use" Command** - Use command **Create a collection for student and Employee** - Collection name

## UNIT II (LECTURE HOURS: 9)

**MongoDB Architecture**

**The Nexus Architecture** - Expressive query language & secondary Indexes, Strong consistency, Enterprise Management and Integrations, Flexible Data Model, Scalability and Performance, Always-On Global Deployments **Install MongoDB**

**Installing MongoDB on windows** - MongoDB package, Create a data directory, Execute MongoDB

**The mongo shell**

**Configure the mongo shell** - Customize the prompt, use an external editor in the mongo shell, change the mongo shell batch size

**Access the mongo shell help** - Command line help, shell help, database help, collection help, cursor help, wrapperobject help

**Write scripts for the mongo shell** - Opening the new connections, difference between interactive and scripted mongo ,scripting

**Data types in the mongo shell** - Types, check types in the mongo shell

## UNIT II TUTORIAL HOUR (LECTURE HOURS: 3)

**Install and run Mongodb**

**Install MongoDB** - Installation

**Run MongoDB server on windows** - Mongod.exe, mongo.exe

## UNIT III (LECTURE HOURS: 10)

**MongoDB CRUD operation**

**Insert Document** - Insert methods

**Query Document** - Nested documents, query on array, embedded documents, project fields to return from query, null or missing fields, iterate a cursor in a mongo shell

**Update Documents** - Update methods **Delete Documents** - Delete methods **Bulk write operations**

**overview** - Db.collection.bulk write()

**Ordered vs unordered operations** - Ordered bulk write, unordered bulk write()

**Bulkwrite method()** - Insertone(), updateone(), updatemany(), replaceone(), deleteone(), deletemany()

**SQL TO MONGODB MAPPING CHART**

**Terminology and concepts** - Terms and concepts of SQL and MongoDB

**Executables** - Executables

**Examples** - Examples

## UNIT III TUTORIAL HOUR (LECTURE HOURS: 2)

**Insert and update methods**

**Insert Methods** - Insertone(),insertmany(),insert()

**Update methods** - Updateone(),updatemany(), replaceone(), update()

## UNIT IV (LECTURE HOURS: 10)

**Text search**

**Text indexes** - Text indexes

**Text search operators** - Query frame work, aggregation frame work

**Text search in the aggregation pipeline** - Restrictions, text score, calculate the total views for articles that contains a word, return results sorted by text search score, match on text score, specify a language for text search

**Indexes**

**Single field indexes** - Create an ascending index on a single field, create an index on an embedded field, create an index on embedded document

**Compound indexes** - Multi key indexes, text indexes

**Hashed indexes** - Index properties, index bulk operations

**Index intersection** - Index prefix intersection, index intersection and compound indexes, index intersection and sort

**Manage indexes** - View existing indexes, remove indexes, modify an index, rebuild indexes

## UNIT IV TUTORIAL HOUR (LECTURE HOURS: 2)

**Measures of index**

**Measure index use** - Indexing strategies

## UNIT V (LECTURE HOURS: 10)

**Replication**

**Replica set primary** - Replica set secondary members

**Replica set arbiter** - Example, security

**ShardingShared cluster** - Shared cluster **Shared keys** - Shared keys **chunks** - chunks**Advantages of sharding** - Read/write, storage capacity, high availability **Shared and non-shared collections** - Shared, non-shared collections **Connecting to a shared cluster** - Mongos, shared cluster

## UNIT V TUTORIAL HOUR (LECTURE HOURS: 2)

**ShardkeYChoosing a shard key** - Restrictions, collection size, shard key cardinality, shard key frequency, monotonically changingshard keys

**Text Books :**

T1. https://docs.mongodb.com/manual/introduction/ | Edition:- | - | -(- )

T2. MongoDB, The definitive guide | Edition:2 | O'Reilly | Kristina chodorow(2013)

T3. Professional NoSQL | Edition:1 | Wiley India Pvt Ltd | Shashank Tiwari(2015)

**Reference Books :**

R2. https://www.edx.org/course/introduction-mongodb-using-mean- | Edition:- | - | -(- ) R1. https://university.mongodb.com/courses/M101P/about