

Master of Computer Application

FLEXILEARN

-Freedom to design your degree



Programme Structure

Curriculum & Scheme of Examination

2015

**AMITY UNIVERSITY
CHHATTISGARH**

RAIPUR

Programme Structure**SECOND SEMESTER**

| Course Code | Course Title | Lecture (L) Hours Per Week | Tutorial (T) Hours Per Week | Practical (P) Hours Per Week | Total Credits |
|-----------------------|--|-----------------------------------|------------------------------------|-------------------------------------|----------------------|
| IFT4201 | Data Structures Using C Language | 3 | - | - | 3 |
| IFT4202 | Operating System | 3 | - | - | 3 |
| IFT4209 | Management Information System | 3 | - | - | 3 |
| IFT4210 | Web Technologies | 3 | - | - | 3 |
| IFT4205 | Computer Oriented Numerical Analysis | 3 | - | - | 3 |
| IFT4206 | Optimization Techniques | 3 | - | - | 3 |
| IFT4207 | Data Structures Using C Language Lab | - | - | 4 | 2 |
| IFT4211 | Web Technologies Lab | - | - | 2 | 1 |
| IFT4212 | Unix Lab | - | - | 2 | 1 |
| Open Electives | | | | | 5 |
| CSS4251 | Corporate Communication | 1 | - | - | 1 |
| BEH4251 | Behavioral Communication and Relationship Management | 1 | - | - | 1 |
| LAN4251 | Foreign Business Language - II | 3 | - | - | 3 |
| LAN4252 | French -II | | | | |
| LAN4253 | German -II | | | | |
| LAN4254 | Spanish -II | | | | |
| LAN4255 | Russian -II | | | | |
| LAN4256 | Chinese -II | | | | |
| LAN4257 | Portuguese -II | | | | |
| LAN4258 | Korean-II | | | | |
| | Japanese-II | | | | |
| | TOTAL | | | | 27 |

SUMMER PROJECT – I**Syllabus – Second Semester****DATA STRUCTURES USING C LANGUAGE****Course Code: IFT4201 Credit Units: 03****Course Objective:**

This course gives the in-depth knowledge to the use, design, and analysis of data structures in computer programs. The very commonly used data structures like arrays, stacks, queues, lists, trees, hashing and graphs will be discussed in detail. Sorting and hashing are important topics in the study of algorithms. They are also closely related to the design of data structures. Several algorithms to implement these techniques are included in the syllabus.

Course Contents:**Module I: Introduction to Data Structures**

Data Types in C Language, Pointers in C, Arrays in C Language, Implementation of Single Dimensional & Multi Dimensional Arrays, Address Calculation in Single and Multi Dimensional Arrays.

Module II: Searching and Sorting Techniques

Insertion Sort, Selection Sort, Merge Sort, Heap Sort, Bubble Sort, Quick Sort, Radix Sort, Hashing, Binary Search, Linear Search, Finding Complexities of Sorting and Searching Algorithms.

Module III: Stacks

Definition & Examples, Representing Stack using an Array, Implementing Push and Pop Operation in a Stack, Infix Postfix & Prefix Expressions, Evaluating a Postfix Expression, Conversion from one form of an expression to another form

Module IV: Queues

Introduction, Queues and its Sequential Representation, Insert –Delete- Print Operations, Circular Queues and their Implementation, Priority Queues with basic operations, De – Que with basic operations.

Module V: Programming with Linked Lists

Introduction, Insertion and deletion of a node to single linked list, Doubly link list with basic operations, List allocating and freeing dynamic variables, Addition of two polynomials list.

Module VI: Programming with Trees

Binary Trees- Operations on Binary Trees, node representation, internal and external nodes ,choosing a b.s.t. representation, threaded binary trees, finding an element in the tree, deleting kth element, tree traversal, constructing a tree, binary search tree- definition. Implementation of B.S.T. Constructing a binary search tree based on its given infix and pre/post fix order, deletion and insertion of a node in B.S.T. Height Balanced Tree. Constructing an AVL Tree, Insertion and Deletion of a node, searching a node

Module VII: Graph and Their Applications

Introduction, Graph Theory Terminology, Sequential Representation of Graph (Adjacency and Path Matrix), Warshall Algorithms, Linked Representation of Graph, Different Operations on Graphs, Traversing A Graph(Dfs, Bfs)., Spanning Trees-Introduction .Representation of Spanning tree, Constructing A Spanning Tree(Prim Algorithm, Krushkal Algorithm)

Module VIII: Hashing

Introduction, Hash Functions, Collision Resolution Techniques.

Examination Scheme:

| Components | CT1 | PR. | ATTD. | EE |
|----------------------|------------|------------|--------------|-----------|
| Weightage (%) | 10 | 15 | 5 | 70 |

Text & References:***Text:***

- Mastering Data Structures Using C. J.B. Dixit, Laxmi Publications.
- Data Structure using C, Yashwant Kanetkar, BPB Publications
- Data Structure using C, Aron M. Tannen Baum, PHI

References:

- Data Structure and Program Design, Robert L Kruse, BPB

OPERATING SYSTEM

Course Code: IFT4202Credit Units: 03

Course Objective:

This course is aimed at providing the basic knowledge of the concepts involved in designing and working of an operating system, how it acts as a resource manager of the system as a whole, how various issues such as Memory conflicts, resource conflicts are resolved by an operating system and a study of various types of operating systems.

Course Contents:

Module I: Introduction

What is an Operating System, Types of Operating Systems, Simple Batch Systems, Multiprogramming Systems, Time-Sharing Systems, Parallel Systems, Distributed Systems, Real-time Systems, Android O.S System.

Module II: Operating System Structures

System Components, System Calls, System Programs, System Structure, Virtual Machines

Module III: Processes Management

Process Concept, Process Scheduling, Operation on processes, Cooperating Processes, Interprocess Communication, Back ground process.

Module IV: CPU Scheduling

Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-Time Scheduling, Algorithm Examination System Models, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery

Module V: Synchronization Construct

Critical section, Problems, semaphore & its implementation.

Module V: Memory Management

Memory Management, Address Space, Memory Allocation Techniques, Swapping, Paging Segmentation with paging, Virtual Memory, Demand Paging, Performance of Demand Paging, Page Replacement, Thrashing, Demand Segmentation

Module VI: File System Interface

File Concept, Access Methods, Directory Structure, Protection, File System Structure, and Allocation Methods.

Module VII: The Unix System Case Study

History, Design Principle, Programmer Interface, User Interface, Process Management, Memory Managements, File management, Interprocess Communication.

Examination Scheme:

| Components | CT1 | A/C/Q | Attd | EE |
|---------------|-----|-------|------|----|
| Weightage (%) | 10 | 15 | 5 | 70 |

Text & References:

Text:

- Operating Systems Concepts, Silberschatz Galvin, Fifth Edition Addition Wesley Publication.
- Modern Operating Systems, A S Tanenbaum, Prentice Hall of India New Delhi, 1995.

References:

- Design of UNIX Operating System, Maurice J. Bauch, Prentice Hall of India.
- Operating Systems Design, Peterson & Galvin

MANAGEMENT INFORMATION SYSTEM

Course Code: IFT4209 Credit Units: 03

Course Objective:

This course will provide the students with an understanding of the principles of information systems technology and its impact on the strategic goals and direction of the organization. They will learn how MIS concepts are applied in business and how information systems can provide solutions to the entire organization.

Course Contents:

Module I: Introduction to MIS

Meaning of MIS, levels of management, function of management, Information and its type, characteristics of information, system and its type, component of MIS, advantages & limitation of MIS, type of information system (DSS, TPS, ES)

Module II: Development of MIS

Introduction, Principles of information, System development, MIS development process, cross life cycle activities.

Module III: Implementation, Evaluation & Maintenance of MIS

Implementation of MIS, Method of Implementing of MIS, Implementation steps of MIS, Evaluation of MIS, Structure for evaluation of MIS, Maintenance of MIS.

Module IV: Concept of Decision Making and MIS

Introduction, Decision Making and manages, Classification of managerial decision, Model for decision making process MIS and Decision Making.

Module V: Advance concept in Information Systems

ERP, Supply chain management, C.R.M, prowerment management, E- commerce.

Examination Scheme:

| Components | CT1 | A/C/Q | Attd | EE |
|---------------|-----|-------|------|----|
| Weightage (%) | 10 | 15 | 5 | 70 |

Text & References:

- MIS by Dharmender kumar
- Management Information Systems, Laudon and Laudon, Prentice Hall International

WEB TECHNOLOGIES

Course Code: IFT4210Credit Units: 03

Course Objective:

This course demonstrate an in-depth understanding of the tools and Web technologies necessary for business application design and development. The course covers client side scripting like HTML, JavaScript and server side scripting like servlets, JSPs. And also XML and web servers and database interfacing.

Course Contents:

Module I

HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets.

Module II

Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script

Module III

XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Module IV

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK
Introspection, Using Bound properties, Bean Info Interface, Constrained properties
Persistence, Customizes, Java Beans API, Introduction to EJB's

Module V

Web Servers and Servlets: Tomcat web server, Introduction to Servlets: Lifecycle of a Servlet, JSDK, The Servlet API, The javax.servelet Package, Reading Servlet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues, Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

Examination Scheme:

| Components | A | CT | S/V/Q | HA | EE |
|---------------|---|----|-------|----|----|
| Weightage (%) | 5 | 10 | 8 | 7 | 70 |

CT: Class Test, HA: Home Assignment, S/V/Q: Seminar/Viva/Quiz, EE: End Semester Examination;
Att: Attendance

TEXT BOOK

1. Deitel & Deitel, Goldberg, "Internet and world wide web - How to Program", Pearson Education Asia, 2001.

REFERENCES

1. Eric Ladd, Jim O' Donnel, "Using HTML 4, XML and JAVA", Prentice Hall of India - QUE, 1999.
2. Aferganatel, "Web Programming: Desktop Management", PHI, 2004.
3. Rajkamal, "Web Technology", Tata McGraw-Hill, 2001.

COMPUTER ORIENTED NUMERICAL ANALYSIS

Course Code: IFT4205

Credit Units: 03

Course Objective:

The course consists of theory and application of numerical approximation techniques. Topics included are numerical error, root-finding, interpolation and polynomial approximation, numerical differentiation and integration, and differential equations.

Course Contents:

Module I: Introduction

Introduction to Numerical Analysis, Application Area of N.A., Numbers and Their Accuracy, Mathematical, preliminaries, Error Analysis, Absolute, Relatives, Truncation and Rounding Error.

Module II: Solution of Linear Algebraic

Introduction to Matrices, Basic Definitions, Matrix Inversion Method, Gauss Jordan Method, Eigen Values, Gauss Seidal Iterative Method

Module III: Solution of Algebraic and Transcendental Equations

Introduction, Bisection Method, Regular Falsi Method, Secant Method, Iteration Method, Newton Raphson Method, Generalized Newton Raphson Method, Rate of Convergence

Module IV: Interpolation and Curve Fitting

Basic Definition, Forward Differences, backward Differences, Central Differences, Newton Forward/Backward/Central Differences Formula, Newton's Divided Difference Formula, LaGrange's Method. Curve Fitting for line and parabola.

Module V: Numerical Difference & Integration

Numerical Differentiation based on Interpolation Formula, Numerical Integration, Trapezoidal Rules, Simpson's 1/3 rule, Simpson's 3/8 Rule, Romberg's Method.

Module VI: Numerical Solution to ODE

Basic Definition, Taylor Series Method, Euler's Method, Modified Euler's Method, Picard's Method, Runge Kutta Method (2nd and 4th Order)

Examination Scheme:

| Components | CT1 | A/C/Q | Attd | EE |
|---------------|-----|-------|------|----|
| Weightage (%) | 10 | 15 | 5 | 70 |

Text & References:

Text:

- V. Raja Raman, Computer Based Numerical Methods, PHI Publication
- SS Sastry, Introductory Methods of Numerical Analysis, PHI

References:

- M.K. Jain & R.K. Jain, Numerical Solutions, New Age Publication

OPTIMIZATION TECHNIQUES

Course Code: IFT4206Credit Units: 03

Course Objective:

Students will learn the tools and techniques of quantitative analysis outlined in the schedule, how and when to apply them, and practice application of those tools. Students completing this goal will be prepared to quantify a variety of policy problems for analysis and decision making. The syllabus includes Linear, Non-linear Programming, Transportation, Decision Theory and Project Management.

Course Contents:

Module I: Linear Programming

Basic Definition, Nature and Significance of OR, feature of OR Approach Application and Scope of OR, General Methods for Solving Or Models. General Structure of Linear Programming, Advantages and Limitations of Linear Programming, Application Areas of Linear Programming. Multiple Solution, Unbounded Solutions, Infeasible Solution

Module II: Simplex Method

Maximization and Minimization Problem, two Phase Method, Big M Method

Module III: Duality in LPP

Dual Linear Programming Problem, Rules for Constructing the Dual from Primal, Feature of Duality

Module V: Transportation Problem

Mathematical Model of Transportation Problem, Transportation Method, North West Corner Method, Linear Cost Method, Vogel's Approximation Method, Unbalanced Supply and Demand, Degeneracy Problem, Alternative Optional Solution, Maximization Transportation Problem, Trans-shipment Problem

Module V: Decision Theory & Decision Tree

Steps in DT Approach, types of Decision Making Environments, Criterion of Optimism and Pessimism, Equally Likely Decision Criterion, Decision Making under Risk, Decision Tree Analysis.

Module VI: Theory of Games

Two Person Zero-Sum Games, Pure Strategies, Game with Saddle Point, Games without Saddle Point, Rule of Dominance, Methods for Solving Problems without Saddle Point.

Module VII: Project Management

Basic Idea of PERT & CRM, Difference between PERT & CPM, PERT/CPM Network Components and Precedence Relationship Critical Path Analysis, Project Scheduling, Project Time-Cost, Trade-Off, Resource Allocation.

Examination Scheme:

| Components | CT1 | A/C/Q | Attd | EE |
|---------------|-----|-------|------|----|
| Weightage (%) | 10 | 15 | 5 | 70 |

Text & References:

Text:

- Operations Research, J K Sharma, Macmillan Publication

References:

- Operations Research, H. A. Taha
- Operations Research, Kanti Swaroop, Macmillan Publication

DATA STRUCTURES USING C LANGUAGE LAB

Course Code: IFT4207Credit Units: 02

Course Contents:

1. Write a program to search an element in a given array using linear search.
2. Write a program to search an element in a given array using binary search.
3. Write a program to sort the elements of an array using Bubble Sort.
4. Write a program to sort the elements of an array using Selection Sort.
5. Write a program to sort the elements of an array using Insertion Sort.
6. Write a program using the concept of iteration and recursion to sort the elements of an array using Quick Sort.
7. Write a C program to implement Heap sort.
8. Write a program to delete an element from the k^{th} element of an array.
9. Write a program to insert an element at the k^{th} element in an array.
10. Write a program to perform the following operations on two given matrices: Addition and Multiplication.
11. Write a program to implement a Stack, show overflow and underflow while performing push and pop operations respectively.
12. Write a program to evaluate a postfix expression.
13. Write a program to implement a queue and show the following: insertion and deletion.
14. Write a program to implement a circular queue and show the following: insertion and deletion.
15. Write a program to implement Linear Linked List and show the following operations: creation, display, insertion, deletion and searching.
16. Write a program to implement a stack using linked list and show the following operations: Push, Pop, and Display.
17. Write a program to implement Doubly Linked List and show the following operations: creation, display, insertion, deletion and searching.
18. Write a program to implement Binary tree and display the contents using preorder, postorder and inorder traversal techniques.
19. Write a program to construct a Binary Search tree and perform the following operations: Insertion and Deletion of a node.
20. Write a program to construct an AVL tree and perform the following operations: insertion, deletion and searching.
21. Write a program to implement Depth First Search using linked representation of graph.
22. Write a program to implement Breadth First Search using linked representation of graph.
23. Write a program to create a minimum spanning tree using Kruskal's algorithm
24. Write a program to create a minimum spanning tree using Prim's algorithm.

Examination Scheme:

| IA | | | | EE | |
|----|----|----|---|----|----|
| A | PR | LR | V | PR | V |
| 5 | 10 | 10 | 5 | 35 | 35 |

Note: IA –Internal Assessment, EE- External Exam, PR- Performance, LR – Lab Record, V – Viva.

WEB TECHNOLOGIES LAB

Course Code:IFT4211 Credit Units: 01

Objective :

To create a fully functional website with mvc architecture. To Develop an online Book store using we can sell books (Ex amazon .com).

Hardware and Software required :

1. A working computer system with either Windows or Linux
2. A web browser either IE or firefox
3. Tomcat web server and Apache web server
4. XML editor like Altova Xml-spy [www.Altova.com/XMLSpy – free] , Stylusstudio , etc.,
5. A database either Mysql or Oracle
6. JVM(Java virtual machine) must be installed on your system
7. BDk(Bea development kit) must be also be installed

Week-1:

Design the following static web pages required for an online book store web site.

1) HOME PAGE:

The static home page must contain three **frames**.

Top frame : Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left frame : At least four links for navigation, which will display the catalogue of respective links.
For e.g.: When you click the link “CSE” the catalogue for CSE Books should be displayed in the Right frame.

Right frame: The *pages to the links in the left frame must be loaded here*. Initially this page contains description of the web site.

| | | | | |
|----------------------------|-----------------------------|--------------|-----------|------|
| Logo | Web Site Name | | | |
| Home | Login | Registration | Catalogue | Cart |
| CSE ECE EEE CIVIL | Description of the Web Site | | | |

Fig 1.1

2) LOGIN PAGE:


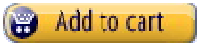



This page looks like below:

| | | | | |
|----------------------------|---|--------------|-----------|------|
| Logo | Web Site Name | | | |
| Home | Login | Registration | Catalogue | Cart |
| CSE ECE EEE CIVIL | <p>Login : <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Submit"/> <input type="button" value="Reset"/></p> | | | |

3) CATALOGUE PAGE:

The catalogue page should contain the details of all the books available in the web site in a table.
The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

| Logo | Web Site Name | | | |
|-------|---|--|-----------|---|
| Home | Login | Registration | Catalogue | Cart |
| CSE |  | Book : XML Bible Author : Winston Publication : Wiely | \$ 40.5 |  |
| ECE | | Book : AI Author : S.Russel Publication : Princeton hall | \$ 63 |  |
| EEE | | Book : Java 2 Author : Watson Publication : BPB publications | \$ 35.5 |  |
| CIVIL | | Book : HTML in 24 hours Author : Sam Peter Publication : Sam publication | \$ 50 |  |

Note: Week 2 contains the remaining pages and their description.

Week-2:

4) CART PAGE:

The cart page contains the details about the books which are added to the cart.
The cart page should look like this:

| | | | | |
|----------------------------|---|----------------------------------|---------------------------|---------------------------------|
| Logo | Web Site Name | | | |
| Home | Login | Registration | Catalogue | Cart |
| CSE ECE EEE CIVIL | Book name Java 2 XML bible | Price \$35.5 \$40.5 | Quantity 2 1 | Amount \$70 \$40.5 |
| | | | Total amount - | \$130.5 |

5) REGISTRATION PAGE:

Create a “*registration form*” with the following fields

- 1) Name (Text field)
- 2) Password (password field)
- 3) E-mail id (text field)
- 4) Phone number (text field)
- 5) Sex (radio button)
- 6) Date of birth (3 select boxes)
- 7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
- 8) Address (text area)

WEEK 3:

VALIDATION:

Write *JavaScript* to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

Note : You can also validate the login page with these parameters.

Week-4:

Design a web page using **CSS (Cascading Style Sheets)** which includes the following:

- 1) Use different font, styles:

In the style definition you define how each selector should work (font, color etc.).
Then, in the body of your pages, you refer to these selectors to activate the styles.

For example:

```
<HTML>
<HEAD>
<style type="text/css">
B.headline { color:red; font-size:22px; font-family:arial; text-
decoration:underline}
</style>

</HEAD>

<BODY>
<b>This is normal bold</b><br>
Selector { cursor:value}
```

For example:

```
<html>
<head>
<style type="text/css">
.xlink { cursor:crosshair}
.hlink{ cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>

<b class="headline">This is headline style bold</b>
</BODY>

</HTML>
```

- 2) Set a background image for both the page and single elements on the page.
You can define the background image for the page like this:

```
BODY { background-image:url(myimage.gif);}
```

- 3) Control the repetition of the image with the background-repeat property.

As background-repeat: repeat

Tiles the image until the entire page is filled, just like an ordinary background image

in

plain HTML.

- 4) Define styles for links as

A:link

A:visited

A:active

A:hover

Example:

```

<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline; color: red;}
</style>

```

5) Work with layers:

For example:

LAYER 1 ON TOP:

```

<div style="position:relative; font-size:50px; z-index:2;">LAYER 1</div>
<div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-
index:1">LAYER 2</div>

```

LAYER 2 ON TOP:

```

<div style="position:relative; font-size:50px; z-index:3;">LAYER 1</div>
<div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-
index:4">LAYER 2</div>

```

6) Add a customized cursor:

Selector { cursor:value }

For example:

```

<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>

```

Week-5:

Write an XML file which will display the Book information which includes the following:

- 1) Title of the book
- 2) Author Name
- 3) ISBN number
- 4) Publisher name
- 5) Edition
- 6) Price

Write a Document Type Definition (DTD) to validate the above XML file.

Display the XML file as follows.

The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns.

Use XML schemas XSL and CSS for the above purpose.

Note: Give at least for 4 books. It should be valid syntactically.

Hint: You can use some xml editors like XML-spy

Week-6:

VISUAL BEANS:

Create a simple visual bean with a area filled with a color.
The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false.
The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the “property window “.

Week-7:

- 1) Install TOMCAT web server and APACHE.
While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.
- 2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root.
Access the pages by using the urls : <http://localhost:4040/rama/books.html> (for tomcat)
<http://localhost:8080/books.html> (for Apache)

Week-8:

User Authentication :

Assume four users user1,user2,user3 and user4 having the passwords pwd1,pwd2,pwd3 and pwd4 respectively. Write a servlet for doing the following.

1. Create a Cookie and add these four user id's and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.

If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display “ You are not an authenticated user “.

Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.

Week-9:

Install a database(Mysql or Oracle).

Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).

Practice 'JDBC' connectivity.

Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

Week-10:

Write a JSP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to week8 instead of cookies).

Week-11:

Create tables in the database which contain the details of items (books in our case like Book name , Price, Quantity, Amount)) of each category. Modify your catalogue page (week 2)in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using JDBC.

Week-12:

HTTP is a stateless protocol. Session is required to maintain the state.

The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time(i.e., from different systems in the LAN using the ip-address instead of localhost). This can be achieved through the use of sessions. Every user will have

his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method `session.invalidate()`).

Modify your catalogue and cart JSP pages to achieve the above mentioned functionality using sessions.

UNIX LAB

Course Code: IFT4212

Credit Units: 01

Unix Programmes

1. Write a Shell Script that takes a search string and filename from the terminal & displays the results.
2. Write a Shell Script that takes pattern and filename as command line arguments and displays the results appropriately i.e. pattern found/pattern not found.
3. Write a Shell Script that accepts only three arguments from the command line. The first argument is the pattern string, the second argument is the filename in which the pattern is to be searched and the third argument is the filename in which the result is to be stored.
4. Write a Shell Script that accepts a filename as a command line argument and finds out if its a regular file or a directory. If its a regular file, then performs various tests to see if it is readable, writeable, executable etc.
5. Write a Shell Script which creates the following menu and prompts for choice from user and runs the chosen command.
 - Today's date
 - Process of user
 - List of files
 - Quit to UNIX
6. Write a Shell Script that computes the factorial of a given number
7. Write a Shell Script that works like a calendar reminding the user of certain things depending on the day of the week.
8. Write a Shell Script that changes the extension of a group of files from txt to doc
9. Write a Shell Script that accepts both filename and a set of patterns as positional parameters to a script.
10. Write a Shell Script which will redirect the output of the date command without the time into a file.
11. Write a Shell Script (using while loop) to execute endlessly (until terminated by user) a loop which displays contents of current directory, disk space status, sleep for 30 seconds and display the users currently logged in on the screen.
12. Write a Shell Script that receives two filenames as arguments. It should check whether content of the two files is same or not. If they are same, second file should be deleted.
13. If a number is input through the keyboard, WASS to calculate sum of its digits.
14. Write a Shell Script that performs a count-down either from 10 (default) or from the value that is entered by the user.
15. Write a Shell Script which takes a command line argument of Kms and by default converts that number into meters. Also provide options to convert km to dm and km to cm.
16. Write a Shell Script using for loop, which displays the message "Welcome to the UNIX System"
17. Write a Shell Script to change the filename of all files in a directory from lower-case to upper-case.
18. Write a Shell Script that examines each file in the current directory. Files whose names end in **old** are moved to a directory named **old files** and files whose names end in **.c** are moved to directory named **cprograms**.
19. Write a Shell Script which searches all files in the given directory (to be taken as command line argument) for the file having the title (to be taken as command line argument), as the first line in the file.
 - a) Display the contents of the searched file.
 - b) In the end, print the file is ###, where
 - ### is small-sized if total no. of lines is <50
 - ### is medium-sized if total no. of lines between 50&100
 - ### is large-sized.
20. Write a shell script which reports names and sizes of all files in a directory (directory would be supplied as an argument to the shell script) whose size is exceeding 1000 bytes. The filenames should be printed in descending order of their sizes. The total number of such files should also be reported.
21. WASS for renaming each file in the directory such that it will have the current shell PID as an extension. The shell script should ensure that the directories do not get renamed.

22. WAP to calculate and print the first *m* Fibonacci numbers.
23. WASS that will receive any number of filenames as arguments. The shell script should check whether such files already exist. If they do, then it should be reported. The files that do not exist should be created in a sub-directory called **mydir**. The shell script should first check whether the sub-directory **mydir** exists in the current directory. If it doesn't exist, then it should be created. If **mydir** already exists, then it should be reported along with the number of files that are currently present in **mydir**.
24. A shell script receives even number of filenames. Suppose four filenames are supplied, then the first file should get copied into second file, the third file should get copied into fourth and so on. If odd number of filenames is supplied then no copying should take place and an error message should be displayed.
25. WASS to identify all zero-byte files in the current directory and delete them. Before proceeding with deletion, the shell script should get a conformation from the user.
26. WASS to compute the **GCD** and **LCM** of two numbers.
27. Two numbers are entered through the keyboard. WAP to find the value of one number raised to the power of another.
28. WASS that prompts the user for the password. The user has maximum of 3 attempts. If the user enters the correct password, the message "Correct Password" is displayed else the message "Wrong Password".
29. WASS that repeatedly asks the user repeatedly for the "Name of the Institution" until the user gives the correct answer.
30. WAP to generate all combinations of 1, 2 and 3 using **for loop**.

Examination Scheme:

| Components | CT1 | PR | Attd | EE |
|---------------|-----|----|------|----|
| Weightage (%) | 10 | 15 | 5 | 70 |