

## Computer Networks

### Course Objectives:

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

1. Build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

### Course Outcomes:

After completing this course the student must demonstrate the knowledge and ability to:

1. Independently understand basic computer network technology.
2. Identify the different types of network topologies and protocols.
3. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.

### Syllabus:

#### UNIT – I :

**Introduction:** OSI overview, TCP/IP and other networks models, Examples of Networks: Novell Networks, Arpanet, Internet, Network Topologies WAN, LAN, MAN.

#### UNIT – II :

**Physical Layer and overview of PL Switching:** Multiplexing: frequency division multiplexing, wave length division multiplexing, synchronous time division multiplexing, statistical time division multiplexing, introduction to switching: Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

#### UNIT – III:

**Data link layer:** Design issues, **Framing:** fixed size framing, variable size framing, flow control, error control, error detection and correction, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, **Elementary Data Link Layer protocols:** simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel.

**Sliding window protocol:** One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC: configuration and transfer modes, frames, control field, point to point protocol (PPP): framing transition phase, multiplexing, multi link PPP.

#### UNIT – IV :

**Random Access:** ALOHA, MAC addresses, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, Controlled Access: Reservation, Polling, Token Passing, Channelization: frequency division multiple access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).

**Network Layer:** Virtual circuit and Datagram subnets-Routing algorithm shortest path routing, Flooding, Hierarchical routing, Broad cast, Multi cast, distance vector routing.

#### UNIT –V :

**IEEE Standards:** – data link layer, physical layer, Manchester encoding, Standard Ethernet: MAC sub layer, physical layer, Fast Ethernet: MAC sub layer, physical layer, IEEE-802.11: Architecture, MAC sub layer, addressing mechanism, frame structure.

#### UNIT –VI :

**Application layer (WWW and HTTP):** ARCHITECTURE : Client (Browser) ,Server ,Uniform Resource Locator  
HTTP: HTTP Transaction, HTTP Operational Model and Client/Server Communication, HTTP Generic Message Format, HTTP Request Message Format, HTTP Response Message Format

**The wireless web :** WAP—The Wireless Application Protocol

**TEXT BOOKS :**

1. Data Communications and Networks – Behrouz A. Forouzan.Third Edition TMH.
2. Computer Networks, 5ed, David Patterson, Elsevier
3. Computer Networks — Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI
4. Computer Networks, Mayank Dave, CENGAGE

**REFERENCES :**

1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education
2. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson

## Data Ware housing and Mining

### Course Objectives:

Students will be enabled to understand and implement classical models and algorithms in data warehousing and data mining. They will learn how to analyze the data, identify the problems, and choose the relevant models and algorithms to apply. They will further be able to assess the strengths and weaknesses of various methods and algorithms and to analyze their behavior.

### Course Outcomes:

1. understand why there is a need for data warehouse in addition to traditional operational database systems;
2. identify components in typical data warehouse architectures;
3. design a data warehouse and understand the process required to construct one;
4. understand why there is a need for data mining and in what ways it is different from traditional statistical techniques;
5. understand the details of different algorithms made available by popular commercial data mining software;
6. solve real data mining problems by using the right tools to find interesting patterns

### Syllabus:

#### UNIT –I:

**Introduction :** What Motivated Data Mining? Why Is It Important, Data Mining—On What Kind of Data, Data Mining Functionalities—What Kinds of Patterns Can Be Mined? Are All of the Patterns Interesting? Classification of Data Mining Systems, Data Mining Task Primitives, Integration of a Data Mining System with a Database or Data Warehouse System, Major Issues in Data Mining. **(Han & Kamber)**

#### UNIT –II:

**Data Pre-processing :** Why Pre-process the Data? Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation. **(Han & Kamber)**

#### UNIT –III:

**Data Warehouse and OLAP Technology: An Overview :** What Is a Data Warehouse? A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, From Data Warehousing to Data Mining. **(Han & Kamber)**

#### UNIT –IV:

**Classification :** Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Working of Decision Tree, building a decision tree, methods for expressing an attribute test conditions, measures for selecting the best split, Algorithm for decision tree induction.

**Model Over fitting:** Due to presence of noise, due to lack of representation samples, evaluating the performance of classifier: holdout method, random sub sampling, cross-validation, bootstrap. **(Tan & Vipin)**

#### UNIT –V

**Association Analysis: Basic Concepts and Algorithms :** Introduction, Frequent Item Set generation, Rule generation, compact representation of frequent item sets, FP-Growth Algorithm. **(Tan & Vipin)**

#### UNIT –VI

**Cluster Analysis: Basic Concepts and Algorithms :** What Is Cluster Analysis? Different Types of Clustering, Different Types of Clusters, K-means, The Basic K-means Algorithm, K-means: Additional Issues, Bisecting K-means, K-means and Different Types of Clusters, Strengths and Weaknesses, K-means as an Optimization Problem, Agglomerative Hierarchical Clustering, Basic Agglomerative Hierarchical Clustering Algorithm, Specific Techniques, DBSCAN, Traditional Density: Center-Based Approach, The DBSCAN Algorithm, Strengths and Weaknesses. **(Tan & Vipin)**

**Text Books :**

1. Introduction to Data Mining : Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson.
2. Data Mining concepts and Techniques, 3/e, Jiawei Han, Michel Kamber, Elsevier.

**Reference Books :**

1. Data Mining Techniques and Applications: An Introduction, Hongbo Du, Cengage Learning.
2. Data Mining : Introductory and Advanced topics : Dunham, Pearson.
3. Data Warehousing Data Mining & OLAP, Alex Berson, Stephen Smith, TMH.
4. Data Mining Techniques, Arun K Pujari, Universities Press.

## Design and Analysis of Algorithms

### Course Objectives:

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

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

### Course Outcomes:

Students who complete the course will have demonstrated the ability to do the following:

- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it.
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
- Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
- Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.
- Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.

### Syllabus:

#### UNIT-I:

Introduction: Algorithm, Psuedo code for expressing algorithms, performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, probabilistic analysis, Amortized analysis.

#### UNIT-II:

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort

#### UNIT-III:

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, spanning trees, Minimum cost spanning trees, Single source shortest path problem.

#### **UNIT-IV:**

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

#### **UNIT-V:**

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

#### **UNIT-VI:**

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution.

#### **TEXT BOOKS:**

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press.
2. Design and Analysis of Algorithms , S Sridhar, Oxford
3. Design and Analysis of Algorithms, Parag Himanshu Dave, Himansu BAlachandra Dave, 2ed, Pearson Education.

#### **REFERENCE BOOKS:**

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin, PEA
3. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd.
4. Algorithm Design, Foundation, Analysis and internet Examples, Michel T Goodrich, Roberto Tamassia, Wiley

## Software Testing

### Course Objectives:

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
5. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
6. To understand software test automation problems and solutions.
7. To learn how to write software testing documents, and communicate with engineers in various forms.
8. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

### Course Outcomes:

By the end of the course, the student should:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems
7. Have an ability to use software testing methods and modern software testing tools for their testing projects.

### Syllabus:

#### UNIT I:

**Software Testing:** Introduction, Evolution, Myths & Facts, Goals, Psychology, Definition, Model for testing, Effective Vs Exhaustive Software Testing.

**Software Testing Terminology and Methodology:** Software Testing Terminology, Software Testing Life Cycle, relating test life cycle to development life cycle, Software Testing Methodology.

#### UNIT II:

**Verification and Validation:** Verification & Validation Activities, Verification, Verification of Requirements, High level and low level designs, How to verify code, Validation

**Dynamic Testing I: Black Box testing techniques:** Boundary Value Analysis, Equivalence class Testing, State Table based testing, Decision table based testing, Cause-Effect Graphing based testing, Error guessing

#### UNIT III:

**Dynamic Testing II: White-Box Testing:** need, Logic coverage criteria, Basis path testing, Graph matrices, Loop testing, data flow testing, mutation testing

**Static Testing:** inspections, Structured Walkthroughs, Technical reviews

#### UNIT IV:

**Validation activities:** Unit testing, Integration Testing,. Function testing, system testing, acceptance testing

**Regression testing:** Progressives Vs regressive testing, Regression testability, Objectives of regression testing, When regression testing done?, Regression testing types, Regression testing techniques

## **UNIT V:**

**Efficient Test Suite Management:** Test case design Why does a test suite grow, Minimizing the test suite and its benefits, test suite prioritization, Types of test case prioritization, prioritization techniques, measuring the effectiveness of a prioritized test suite

**Software Quality Management:** Software Quality metrics, SQA models

Debugging: process, techniques, correcting bugs, Basics of testing management tools, test link and Jira

## **UNIT VI:**

**Automation and Testing Tools:** need for automation, categorization of testing tools, selection of testing tools, Cost incurred, Guidelines for automated testing, overview of some commercial testing tools.

**Testing Object Oriented Software:** basics, Object oriented testing

**Testing Web based Systems:** Challenges in testing for web based software, quality aspects, web engineering, testing of web based systems, Testing mobile systems

### **Text Books:**

1. Software Testing, Principles and Practices, Naresh Chauhan, Oxford
2. Foundations of Software testing, Aditya P Mathur, 2ed, Pearson
3. Software Testing- Yogesh Singh, CAMBRIDGE

### **Reference books:**

1. *Software testing techniques - Boris Beizer, International Thomson computer press, second edition.*
2. Software Testing, Principles, techniques and Tools, M G Limaye, TMH
3. Effective Methods for Software testing, William E Perry, 3ed, Wiley



## Web Technologies

### Course Objectives:

This course is designed to introduce students with no programming experience to the programming languages and techniques associated with the World Wide Web. The course will introduce web-based media-rich programming tools for creating interactive web pages.

### Course Outcomes:

1. Analyze a web page and identify its elements and attributes.
2. Create web pages using XHTML and Cascading Styles sheets.
3. Build dynamic web pages .
4. Build web applications using PHP.
5. Programming through PERL and Ruby
6. write simple client-side scripts using AJAX

### Syllabus:

#### UNIT-I:

HTML tags, Lists, Tables, Images, forms, Frames. Cascading style sheets. Introduction to Java script. Objects in Java Script. Dynamic HTML with Java Script

#### UNIT-II:

**Working with XML:** Document type Definition, XML schemas, Document object model, XSLT, DOM and SAX.

#### UNIT-III:

**AJAX A New Approach:** Introduction to AJAX, Integrating PHP and AJAX. Consuming WEB services in AJAX: (SOAP, WSDL,UDDI)

#### UNIT-IV:

**PHP Programming: Introducing PHP:** Creating PHP script, Running PHP script. **Working with variables and constants:** Using variables,Using constants,Data types,Operators.**Controlling program flow:** Conditional statements,Control statements,Arrays,functions.Working with forms and Databases such as mySql, Oracle, SQL Sever.

#### UNIT-V:

Introduction to PERL, Perl language elements, Interface with CGI- A form to mail program, Simple page search

#### UNIT-VI:

Introduction to Ruby, variables, types, simple I/O, Control, Arrays, Hashes, Methods, Classes, Iterators, Pattern Matching, Practical Web Applications

### Text Books:

1. Programming the World Wide Web, Robet W Sebesta, 7ed, Pearson.
2. Web Technologies, Uttam K Roy, Oxford
3. The Web Warrior Guide to Web Programming, Bai, Ekedahl, Farrelll, Gosselin, Zak, Karparhi, MacIntyre, Morrissey, Cengage

### Reference Books:

1. Ruby on Rails Up and Running, Lightning fast Web development, Bruce Tate, Curt Hibbs, Oreilly ( 2006)
2. Programming Perl, 4ed, Tom Christiansen, Jonathan Orwant, Oreilly (2012)
3. Web Technologies, HTML< JavaScript, PHP, Java, JSP, XML and AJAX, Black book, Dream Tech.
4. An Introduction to Web Design, Programming, Paul S Wang, Sanda S Katila, Cengage

## Computer Networks & Network Programming Lab

### Objectives:

- To teach students practice orientation of networking concepts
- To teach students various forms of IPC through Unix and socket Programming

### PART – A

1. Implement the data link layer framing methods such as character stuffing and bit stuffing.
2. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
3. Implement Dijkstra's algorithm to compute the Shortest path thru a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm
5. Take an example subnet of hosts. Obtain broadcast tree for it.

### PART – B

1. Implement the following forms of IPC.  
a) Pipes                      b) FIFO
2. Implement file transfer using Message Queue form of IPC
3. Write a programme to create an integer variable using shared memory concept and increment the variable
4. simultaneously by two processes. Use semaphores to avoid race conditions
5. Design TCP iterative Client and server application to reverse the given input sentence
6. Design TCP iterative Client and server application to reverse the given input sentence
7. Design TCP client and server application to transfer file
8. Design a TCP concurrent server to convert a given text into upper case using multiplexing system call "select"
9. Design a TCP concurrent server to echo given set of sentences using poll functions
10. Design UDP Client and server application to reverse the given input sentence
11. Design UDP Client server to transfer a file
12. Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
13. Design a RPC application to add and subtract a given pair of integers

## Software Testing Lab

### Lab Assignments

#### Problem Statement 01

Consider an automated banking application. The user can dial the bank from a personal computer, provide a six-digit password, and follow with a series of keyword commands that activate the banking function. The software for the application accepts data in the following form:

Area Code	Blank or three-digit number
Prefix	Three-digit number, not beginning with 0 or 1
Suffix	Four-digit number
Password	Six-character alphanumeric
Commands	"Check status", "Deposit", "Withdrawal"

Design adhoc test cases to test the system

#### Problem Statement 02

Consider an automated banking application. The user can dial the bank from a personal computer, provide a six-digit password, and follow with a series of keyword commands that activate the banking function. The software for the application accepts data in the following form:

Area Code	Blank or three-digit number
Prefix	Three-digit number, not beginning with 0 or 1
Suffix	Four-digit number
Password	Six-character alphanumeric
Commands	"Check status", "Deposit", "Withdrawal"

Design the test cases to test the system using following Black Box testing technique:

BVA, Worst BVA, Robust BVA, Robust Worst BVA  
Equivalence class testing (Input/Output domain)

#### Problem Statement 03

Consider an application that is required to validate a number according to the following simple rules:

1. A number can start with an optional sign.
2. The optional sign can be followed by any number of digits.
3. The digits can be optionally followed by a decimal point, represented by a period.
4. If there is a decimal point, then there should be two digits after the decimal.
5. Any number-whether or not it has a decimal point, should be terminated a blank.
6. A number can start with an optional sign.
7. The optional sign can be followed by any number of digits.
8. The digits can be optionally followed by a decimal point, represented by a period.
9. If there is a decimal point, then there should be two digits after the decimal.

10. Any number-whether or not it has a decimal point, should be terminated a blank. Generate test cases to test valid and invalid numbers.

**(HINT)** Use Decision table and cause-effect graph to generate test cases.

#### Problem Statement 04

Generate test cases using Black box testing technique to Calculate Standard Deduction on Taxable Income. The standard deduction is higher for tax payers who are 65 or older or blind. Use the method given below to calculate tax.

1. The first factor that determines the standard deduction is the filing status. The basic standard deduction for the various filing status are:

Single	\$4,750
Married, filing a joint return	\$9,500
Married, filing a separate return	\$7,000

2. If a married couple is filing separate returns and one spouse is not taking standard Deduction, the other spouse also is not eligible for standard deduction.

3. An additional \$1,000 is allowed as standard deduction, if either the filer is 65 yrs or the spouse is 65 yrs or older (the latter case applicable when the filing status is “Married” and filing “joint”).

4. An additional \$1,000 is allowed as standard deduction, if either the filer is blind or the spouse is blind (the latter case applicable when the filing status is “married” and filing “joint”).

#### **(HINT):**

From the above description, it is clear that the calculation of standard deduction depends on the following 3 factors:

1. Status of filing of the filer
2. Age of the filer
3. Whether the filer is blind or not

In addition, in certain cases, the following additional factors also come into play in calculating the standard deduction.

1. Whether spouse has claimed standard deduction
2. Whether spouse is blind
3. Whether the spouse is more than 65 years old

#### Problem Statement 05

Consider the following program segment:

```
1. int max (int i, int j, int k)
2. {
3.   int max;
4.   if (i>j) then
5.     if (i>k) then max=i;
6.   else max=k;
7.   else if (j > k) max=j
8.   else max=k
9.   return (max);
10. }
```

- a) Draw the control flow graph for this program segment
- b) Determine the cyclomatic complexity for this program
- c) Determine the independent paths

#### Problem Statement 06

Source code of simple insertion sort implementation using array in ascending order in c programming language

```

#include<stdio.h>
int main(){
int i,j,s,temp,a[20];
Printf("Enter total elements: "); Scanf ("%d",&s);
printf("Enter %d elements: ",s); for(i=0;i<s;i++) scanf("%d",&a[i]); for(i=1;i<s;i++){
temp=a[i]; j=i-1; while((temp<a[j])&&(j>=0)){ a[j+1]=a[j];
j=j-1;
}
a[j+1]=temp;
}
printf("After sorting: ");
for(i=0;i<s;i++)
printf(" %d",a[i]);
return 0;
}

```

HINT: for loop is represented as while loop

- Draw the program graph for given program segment
- Determine the DD path graph
- Determine the independent paths
- Generate the test cases for each independent path

### Problem Statement 07

Consider a system having an FSM for a stack having the following states and transitions:

#### States

Initial: Before creation

Empty: Number of elements = 0

Holding: Number of elements > 0, but less than the maximum capacity

Full: Number elements = maximum

Final: After destruction

Initial to Empty: Create

Empty to Holding, Empty to Full, Holding to Holding, Holding to Full: Add

Empty to Final, Full to Final, Holding to Final: Destroy

Holding to Empty, Full to Holding, Full to Empty: Delete

Design test cases for this FSM using state table-based testing.

### Problem Statement 08

Given the following fragment of code, how many tests are required for 100% decision coverage? Give the test cases.

```

if width > length
then biggest_dimension = width
if height > width
then biggest_dimension = height
end_if
else if biggest_dimension = length
then if height > length
then biggest_dimension = height
end_if
end_if
end_if

```

**Hint** 04 test cases

### Problem Statement 09

Given the following code, how much minimum number of test cases is required for full statement and branch coverage?

```

read p read q
if p+q > 100
then print "Large"
endif

```

```
if p > 50
then print "p Large" endif
```

**Hint** 1 test for statement coverage, 2 for branch coverage

### Problem Statement 10

Consider a program to input two numbers and print them in ascending order given below. Find all du paths and identify those du-paths that are not feasible. Also find all dc paths and generate the test cases for all paths (dc paths and non dc paths).

```
#include<stdio.h>
#include<conio.h>
1. void main ()
2. {
3 int a, b, t;
4. Clrscr ();
5. Printf ("Enter first number");
6. scanf ("%d",&a);
7. printf("Enter second number");
8. scanf("%d",&b);
9. if (a<b){
10. t=a;
11 a=b;
12 b=t;
13}
14. printf ("%d %d", a, b);
15 getch ();
}
```

### Problem Statement 11

Consider the above program and generate possible program slices for all variables. Design at least one test case from every slice.

### Problem Statement 12

Consider the code to arrange the nos. in ascending order. Generate the test cases for relational coverage, loop coverage and path testing. Check the adequacy of the test cases through mutation testing and also compute the mutation score for each.

```
i = 0;
n=4; //N-Number of nodes present in the graph
While (i<n-1) do j = i + 1;
While (j<n) do
if A[i]<A[j] then swap (A[i], A[j]); end do;
i=i+1;
end do
```

III Year – II SEMESTER

<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>3</b>	<b>2</b>

## Web Technologies Lab

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

1) **HOME PAGE:**

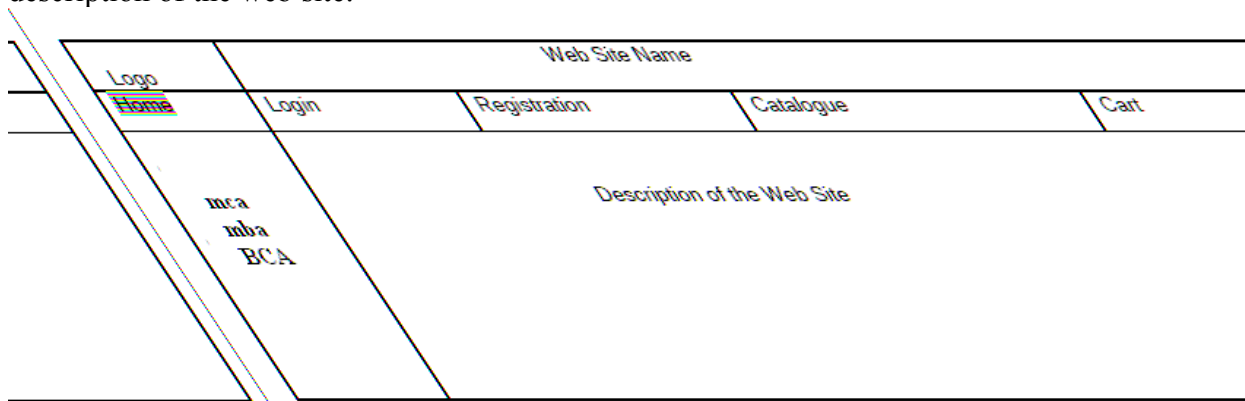
The static home page must contain three **frames**.

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).

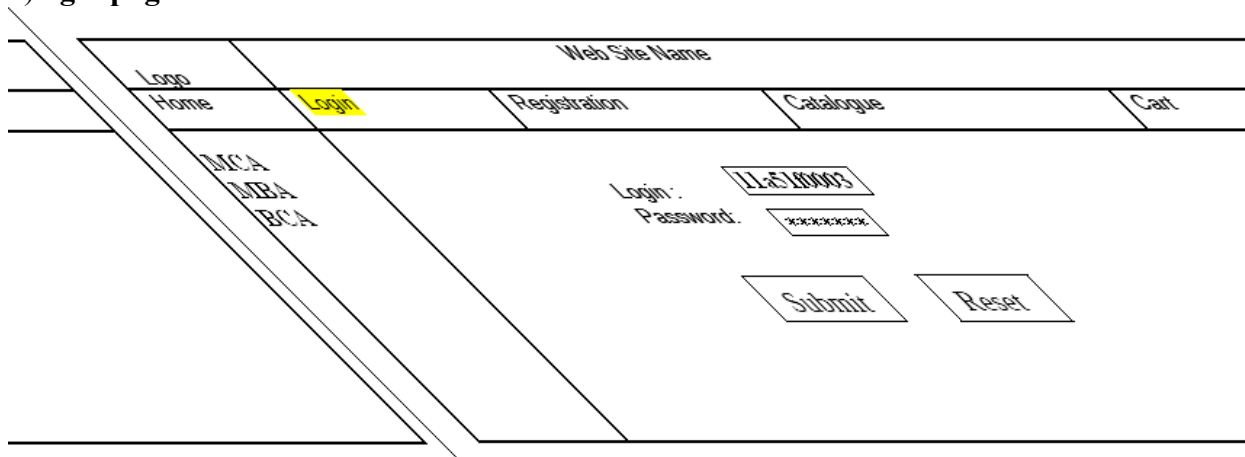
Left frame : At least four links for navigation, which will display the catalogue of respective links.

For e.g.: When you click the link “**MCA**” the catalogue for MCA 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.



## 2)login page











## 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.

Web Site Name					
Logo					
Home	Login	Registration	Catalogue	Cart	
MCA	   	Book : XML Bible Author : Winston Publication : Wiley	\$ 40.5		
MEB		Book : AI Author : S.Russel Publication : Princeton hall	\$ 63		
BCA		Book : Java 2 Author : Watson Publication : BPB publications	\$ 35.5		
		Book : HTML in 24 hours Author : Sam Peter Publication : Sam	\$ 50		

#### 4. 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)

5. 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

6. 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.

7. Write Ruby program reads a number and calculates the factorial value of it and prints the same.

8. Write a Ruby program which counts number of lines in a text file using its regular expressions facility.

9. Write a Ruby program that uses iterator to find out the length of a string.

10. Write simple Ruby programs that uses arrays in Ruby.

11. Write programs which uses associative arrays concept of Ruby.

12. Write Ruby program which uses Math module to find area of a triangle.

13. Write Ruby program which uses tk module to display a window



14. Define complex class in Ruby and do write methods to carry operations on complex objects.
15. Write a program which illustrates the use of associative arrays in perl.
16. Write perl program takes a set names along the command line and prints whether they are regular files or special files
17. Write a perl program to implement UNIX `passwd' program
18. An example perl program to connect to a MySQL database table and executing simple commands.
19. Example PHP program for cotactus page.

## **20. User Authentication :**

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

21. Example PHP program for registering users of a website and login.

22. 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).

Write a PHP program 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).

## **23. Write a PHP 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).

## **24. 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 PHP**

## **25. 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 PHP pages to achieve the above mentioned functionality using sessions.

**INTELLECTUAL PROPERTY RIGHTS AND PATENTS -  
II**

**UNIT - I**

Intellectual Property Law Basics – Types of Intellectual Property – Agencies responsible for Intellectual Property Registration - Cyber crime and E-commerce – International Aspects of Computer and Online Crime

**UNIT - II**

Introduction to Patent Law – Rights and Limitations – Rights under Patent Law –Patent requirements - Ownership - Transfer - Patents Application Process – Patent Infringement - Patent Litigation - International Patent Law – Double Patenting – Patent Searching – Patent Law Treaty - New developments in Patent Law - Invention Developers and Promoters

**UNIT - III**

Introduction to Transactional Law: Creating Wealth and Managing Risk – The Employment Relationship in the Internet and Tech Sector – Contact for the Internet and Tech Sector - Business Assets in Information Age – Symbol and Trademark – Trolls and Landmines and other Metaphors

**UNIT - IV**

Regulatory , Compliance and Liability Issues – State Privacy Law - Data Security – Privacy issues - Controlling Over use or Misuse of Intellectual Property Rights

**BOOKS:**

1. Deborah E.Bouchoux: “Intellectual Property”. Cengage learning , New Delhi
2. Kompal Bansal & Parishit Bansal "Fundamentals of IPR for Engineers", BS Publications (Press)
3. Cyber Law. Texts & Cases, South-Western's Special Topics Collections
4. Prabhuddha Ganguli: ‘ Intellectual Property Rights’ Tata Mc-Graw – Hill, New Delhi
5. Richard Stim: "Intellectual Property", Cengage Learning, New Delhi.
6. R. Radha Krishnan, S. Balasubramanian: "Intellectual Property Rights", Excel Books. New Delhi.
7. M.Ashok Kumar and Mohd.Iqbal Ali: “Intellectual Property Right” Serials Pub.

## Cryptography and Network Security

### Course objectives:

The main objective of this course is to teach students to understand and how to address various software security problems in a secure and controlled environment. During this course the students will gain knowledge (both theoretical and practical) in various kinds of software security problems, and techniques that could be used to protect the software from security threats. The students will also learn to understand the “modus operandi” of adversaries; which could be used for increasing software dependability.

### Course outcomes:

1. be able to individually reason about software security problems and protection techniques on both an abstract and a more technically advanced level.
2. be able to individually explain how software exploitation techniques, used by adversaries, function and how to protect against them.

### Syllabus:

#### UNIT I : Classical Encryption Techniques

**Objectives:** *The Objectives of this unit is to present an overview of the main concepts of cryptography, understand the threats & attacks, understand ethical hacking.*

**Introduction:** Security attacks, services & mechanisms, Symmetric Cipher Model, Substitution Techniques, Transportation Techniques, Cyber threats and their defense( Phishing Defensive measures, web based attacks, SQL injection & Defense techniques)(TEXT BOOK 2), Buffer overflow & format string vulnerabilities, TCP session hijacking(ARP attacks, route table modification) UDP hijacking ( man-in-the-middle attacks)(TEXT BOOK 3).

#### UNIT II: Block Ciphers & Symmetric Key Cryptography

**Objectives:** *The Objectives of this unit is to understand the difference between stream ciphers & block ciphers, present an overview of the Feistel Cipher and explain the encryption and decryption, present an overview of DES, Triple DES, Blowfish, IDEA.*

Traditional Block Cipher Structure, DES, Block Cipher Design Principles, AES-Structure, Transformation functions, Key Expansion, Blowfish, CAST-128, IDEA, Block Cipher Modes of Operations

#### UNIT III: Number Theory & Asymmetric Key Cryptography

**Objectives:** *Presents the basic principles of public key cryptography, Distinct uses of public key cryptosystems*

**Number Theory:** Prime and Relatively Prime Numbers, Modular Arithmetic, Fermat’s and Euler’s Theorems, The Chinese Remainder theorem, Discrete logarithms.

**Public Key Cryptography:** Principles, public key cryptography algorithms, RSA Algorithms, Diffie Hellman Key Exchange, Elgamal encryption & decryption, Elliptic Curve Cryptography.

#### UNIT IV : Cryptographic Hash Functions & Digital Signatures

**Objectives:** *Present overview of the basic structure of cryptographic functions, Message Authentication Codes, Understand the operation of SHA-512, HMAC, Digital Signature*

Application of Cryptographic hash Functions, Requirements & Security, Secure Hash Algorithm, Message Authentication Functions, Requirements & Security, HMAC & CMAC. Digital Signatures, NIST Digital Signature Algorithm. Key management & distribution.

#### UNIT V: User Authentication, Transport Layer Security & Email Security

**Objectives:** *Present an overview of techniques for remote user authentication, Kerberos, Summarize Web Security threats and Web traffic security approaches, overview of SSL & TLS. Present an overview of electronic mail security.*

**User Authentication:** Remote user authentication principles, Kerberos

**Transport Level Security:** Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS), Secure Shell (SSH)

**Electronic Mail Security:** Pretty Good Privacy (PGP) and S/MIME.

## **UNIT VI: IP Security & Intrusion Detection Systems**

**Objectives:** *Provide an overview of IP Security, concept of security association, Intrusion Detection Techniques*

**IP Security:** IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management.

**Intrusion detection:** Overview, Approaches for IDS/IPS, Signature based IDS, Host based IDS/IPS. (TEXT BOOK 2)

### **TEXT BOOKS:**

1. Cryptography & Network Security: Principles and Practices, William Stallings, PEA, Sixth edition.
2. Introduction to Computer Networks & Cyber Security, Chwan Hwa Wu, J. David Irwin, CRC press
3. Hack Proofing your Network, Russell, Kaminsky, Forest Puppy, Wiley Dreamtech.

### **REFERENCE BOOKS:**

1. Everyday Cryptography, Fundamental Principles & Applications, Keith Martin, Oxford
2. Network Security & Cryptography, Bernard Menezes, Cengage, 2010