

**Savitribai Phule Pune University, Pune**  
 Second Year B.C.A.(Science)  
 Semester III  
 To be implemented Academic year 2017-18

**Course Code: BCA-301**  
**Total Contact Hours: 48 hrs.**  
**(60 Lectures)**

**Course Title: Data Structure**  
**Total Credits: 04 Total Marks: 100**

**Teaching Scheme: Theory - 05 Lectures/ Week**

**Course Objective:**

- To study the various structures or methods of organizing data in computer's memory and efficiently implement them.

**Prerequisite:** Knowledge of C programming language

Unit No.	Content	No. of lecture
1	<b>Introduction to data structure</b> 1.1 Data type and data object 1.2 Abstract Data Type (ADT) 1.3 Type of data structure 1.4 Algorithm analysis 1.4.1 Space and Time complexity 1.4.2 Asymptotic notation – BigO, OmegaΩ 1.4.3 Simple algorithms and its complexity as examples	4
2	<b>Array as a data structure</b> 2.1 Array representation- Row and column major 2.2 Sorting techniques with time complexity: Bubble sort, Insertion sort, Merge sort, Quick sort 2.3 Searching techniques with time Complexity: Linear search, Binary search	8
3	<b>Linked List</b> 3.1 Introduction 3.2 Representation -Static &Dynamic 3.3 Types of linked lists- singly, doubly, circular 3.4. Operations-create, display, insert, delete, reverse, search, sort, concat, merge 3.5 Applications –single variable polynomial manipulation	12
4	<b>Stack</b> 4.1 Introduction 4.2 Representation -Static &Dynamic 4.3 Operations– push, pop 4.4 Applications – Recursion, Infix to postfix, Infix to prefix, postfix evaluation	9
5	<b>Queue</b> 5.1 Introduction 5.2 Representation -Static &Dynamic 5.3 Operations– insert, delete 5.4 Circular queue 5.5 Priority queue 5.6 Applications	8

6	<b>Tree</b> 6.1 Introduction 6.2 Tree terminologies- all definitions, root, leaf, level, height, depth 6.3 Binary tree 6.3.1 Types 6.3.2 Types of Traversal– Preorder, Inorder, Postorder 6.3.3 Representation – Static & Dynamic 6.4 Binary Search Tree (BST)	13
7	<b>Graph</b> 7.1 Introduction 7.2 Graph terminologies 7.3 Representation – Adjacency matrix, Adjacency list, Adjacency multilist 7.4 Traversal– DFS, BFS 7.5 Applications AOV network – Topological sort AOE network – Critical path	6

**Reference Books:**

1. Fundamentals of Data Structures-Horowitz Sahani (Galgotia)
2. Introduction to Data Structures using C-Ashok Kamthane
3. Data Structures using C-Bandopadhyay & Dey(Pearson)
4. Data Structures using C-By Srivastava BPB Publication.

**Course Code: BCA302      Course Title: Advanced Relational Database Management System**  
**Total Contact Hours: 48 hrs.      Total Credits: 04      Total Marks: 100**  
**(60 Lectures)**

**Teaching Scheme: Theory-05 Lectures/ Week**

**Course Objective:**

- To study fundamental concepts of RDBMS (PL/Pgsql)
- To study database management operations
- To study data security and its importance
- To study client server architecture

Unit No.	Content	No. of Lectures
1	<p><b>Relational Database Design</b></p> <p>1.1. PL/Pgsql: Language structure            1.2. Controlling the program flow, conditional statements, loops            1.3. Views            1.4. Functions            1.5. Handling errors and exceptions            1.6. Cursors            1.7. Triggers</p>	16
2	<p><b>Transaction Concepts and Concurrency Control</b></p> <p>2.1 Transaction, properties of transaction, states of transactions            2.2 Concurrent execution of transactions and conflicting operations            2.3 Schedules, types of schedules, concept of Serializability, precedence graph for Serializability            2.4 Ensuring Serializability by locks, different lock modes, 2PL and its variations            2.5 Multigranularity locking protocol            2.6 Basic timestamp method for concurrency, Thomas Write Rule            2.7 Locks with multiple granularity, dynamic database concurrency (Phantom Problem)            2.8 Timestamps versus locking            2.9 Optimistic concurrency control algorithm, multi version concurrency control            2.10 Deadlock handling methods                2.10.1 Detection and Recovery (Wait for graph).                2.10.2 Prevention algorithms (Wound-wait, Wait-die)</p>	16

3	<b>Crash Recovery</b> 3.1 Transaction Failure classification 3.2 Recovery concepts 3.3 Checkpoints 3.4 Recovery with concurrent transactions (Rollback, checkpoints, commit) 3.5 Log base recovery techniques (Deferred and Immediate update) 3.6 Buffer Management 3.7 Relationship between Recovery management and Buffer management 3.8 Aries algorithm 3.9 Database backup and recovery from catastrophic failure 3.10 Shadow paging	16
4	<b>Database Security</b> 4.1 Introduction to database security concepts 4.2 Methods for database security 4.3 Discretionary access control method 4.4 Mandatory access control and role based access control for multilevel security 4.5 Use of views in security enforcement 4.6 Overview of encryption technique for security 4.7 Statistical database security	6
5	<b>Client-Server Technology</b> 5.1 Client-server computing 5.2 Evolution of Client-Server information systems 5.3 Client– Server Architecture benefits 5.4 Client Server Architecture (2 tier and 3 tier) 5.5 Components, Principles, Client Components 5.6 Communication middleware components 5.7 Database middleware components 5.8 Client Server Databases	6

**Reference Books:**

1. Database System Concepts – Avi Silberschatz, Henry F. Korth, S. Sudarshan, 6th edition- McGraw-Hill
2. Fundamentals of Database Systems-RamezElmasri, ShamkantNavathe, 5th edition–Pearson.
3. Practical Postgresql, JoshuaD. Drake, John C Worsley, O’Reilly Publications.
4. Database Management Systems -Raghu Ramakrishnan, 3rdEdition, Tata McGraw Hill
5. Database Management System- Bipin Desai

**Course Code: BCA-303**  
**Total Contact Hours: 48 hrs.**  
**(60 Lectures)**

**Course Title: Software Engineering**  
**Total Credits: 04**      **Total Marks: 100**

**Teaching Scheme: Theory-05 Lectures/ Week**

**Course Objectives:**

- The Objective of this course is to understand system concepts, to know about software engineering and its application in Software development

Prerequisite: ERD

<b>Unit No.</b>	<b>Content</b>	<b>No. of Lectures</b>
1	<b>Introduction to System Concepts</b> 1.1 Definition 1.2 Basic Components 1.3 Elements of the system 1.4 System Components 1.5 Types of System	<b>2</b>
2	<b>Introduction to Software Engineering</b> 2.1 Definition of Software 2.2 Characteristics of Software 2.3 Software Application Domains 2.4 Definition of Software Engineering 2.5 Need for software Engineering 2.6 Mc Call's Quality factors 2.7 The Software Process 2.8 Software Engineering Practice	<b>6</b>
3	<b>Software Development Life Cycle(SDLC)and Methodologies</b> 3.1 Introduction 3.2 Activities of SDLC 3.3 A Generic Process Model 3.4 Prescriptive Process models 3.4.1 Waterfall Model 3.4.2 Incremental Process Models 3.4.3 Evolutionary process Models (Prototyping and Spiral Model) 3.5 Concurrent Models, Types	<b>12</b>
4	<b>Requirement Engineering</b> 4.1 Introduction 4.2 Requirement Engineering Tasks 4.3 Establishing Groundwork for understanding of Software Requirement 4.4 Requirement Gathering 4.5 Feasibility study 4.6 Fact Finding Techniques	<b>8</b>

5	<b>Analysis And Design Tools</b> 5.1 Decision Tree and Decision Table 5.2 Data Flow Diagrams (DFD) 5.3 Data Dictionary 5.3.1 Elements of DD 5.3.2 Advantages of DD 5.4 Input and Output Design 5.5 PseudoCode 5.6 Case Studies on above topics	<b>12</b>
6	<b>Software Testing</b> 6.1 Definition 6.2 Verification And Validation 6.3 Black box and White-Box Testing 6.4 Unit Testing 6.5 Integration Testing 6.6 System Testing 6.6.1 Performance Testing 6.6.2 Stress Testing 6.7 Smoke Testing 6.8 User Acceptance Testing	<b>8</b>
7	<b>Maintenance and Reengineering</b> 7.1 Maintenance definition and types 7.2 Software reengineering 7.3 Reverse Engineering 7.4 Restructuring and forward Engineering.	<b>4</b>
8	<b>Agile Development</b> 8.1 Agility 8.2 Agile Process 8.2.1 Principles 8.2.2 The Politics Of Agile Development 8.2.3 Human Factors 8.3 Extreme Programming(XP) 8.4 Adaptive Software Development(ASD) 8.5 Scrum 8.6 Dynamic System Development Model (DSDM)	<b>8</b>

**Reference Books:**

1. Software Engineering : A Practitioner's Approach- Roger S. Pressman, McGraw hill International Editions 2010(Seventh Edition)
2. System Analysis, Design and Introduction to Software Engineering (SADSE) - S. Parthasarthy, B.W. Khalkar
3. Analysis and Design of Information Systems(Second Edition) - James A. Senn, McGraw Hill
4. System Analysis and Design- Elias Awad, Galgotia Publication, Second Edition
5. Fundamentals of Software Engineering- Rajib Mall, PHI Publication, Fourth Edition

**Course Code: BCA-304**  
**Total Contact Hours: 48 hrs.**  
**(60 Lectures)**

**Course Title: Introduction to Computer Network**  
**Total Credits: 04**      **Total Marks: 100**

**Teaching Scheme: Theory-05 Lectures/ Week**

**Course Objectives:**

- To prepare students with basic networking concepts: data communication, protocols and standards, various topologies and applications of network.

<b>Unit No.</b>	<b>Content</b>	<b>No. of Lectures</b>
1	<b>An Introduction to Networks, Network Topologies and Types</b> 1.1 Data communications and representation 1.2 Information Exchange, Sharing, preserving &protecting 1.3 Hardware and Software Resource Sharing 1.4ComputerNetworks-Goals and applications –Business Application, Home Application, Mobile User, Social Issues 1.5 Network Hardware -Broadcast and point-to-point 1.6 Clients, Servers, Peers based and Hybrid Networks 1.7 Network Topologies (Bus, Star, Ring, Star Bus, Mesh) 1.8 Network Types- LAN, MAN, WAN, PAN, Wireless Networks, Home Networks, Internetwork	12
2	<b>Network Models</b> 2.1. Introduction to OSI Model with all layers 2.2 TCP/IP Protocol Suite 2.3 Addressing-Physical, Logical and Port addresses	4
3	<b>Overview of Transmission Media</b> 3.1 Introduction to Guided Transmission Media : Twisted pair cable–UTP Vs STP, Categories connectors &applications, Coaxial cable – standards, Connectors & applications, Fiber Optic cable–propagation modes, connectors & applications 3.2 Unguided Media – Wireless- Radio Waves,- Microwaves, Infrared, Satellite Communication. 3.3 Types of cabling and Networking Tool: CAT5 andCAT6 Cable Color Code, Cross over Cabling and Straight through Cable.	5

4	<p><b>Physical Layer</b></p> <p>4.1 Analog and Digital data, Analog and Digital signals, Periodic &amp; Non-periodic signals, Digital Signals-Bit rate, bit length, baseband Transmission.</p> <p>4.2 Transmission Impairments—attenuation, distortion and noise, Data Rate Limits— Noiseless channel: Nyquist’s bit rate, noisy channel : Shannon’s law</p> <p>4.3 Performance of the Network Bandwidth, Throughput, Latency(Delay), Bandwidth – Delay Product, Jitter</p> <p>4.4 Line Coding Characteristics, Line Coding Schemes—Unipolar -NRZ, Polar-NRZ-I, NRZ-L, RZ, Manchester and Differential Manchester, Problems</p> <p>4.5 Transmission Modes, Parallel Transmission and Serial Transmission— Asynchronous and Synchronous and Isochronous</p> <p>4.6 Trunks &amp; Multiplexing FDM and TDM</p> <p>4.7 Switching-Circuit Switching, Message Switching and Packet Switching</p> <p>4.8 Physical Layer Devices: Repeaters, Hubs-active hub Passive hub</p>	14
5	<p><b>Data Link Layer</b></p> <p>5.1 Design Issues—Services provided to the Network Layer, Framing— Concept, Methods—Character Count, Flag bytes with Byte Stuffing, Starting &amp; ending Flags with Bit Stuffing and Physical Layer Coding Violations, Flow and Error Control</p> <p>5.2 Error detection code CRC</p> <p>5.3 Data Link Layer Devices-Bridges Filtering, Transparent Bridges, spanning tree and Source, Routing Bridges, Bridges Connecting different LAN’s.</p> <p>5.4 Random Access Protocols ALOHA— pure and slotted, CSMA- 1- persistent, p-persistent and non-persistent CSMA/CD, CSMA/CA</p> <p>5.5 Controlled Access Reservation, Polling and Token Passing</p> <p>5.6 Channelization FDMA, TDMA and CDMA-Analogy, Idea, Chips, Data Representation, Encoding and Decoding, Signal Level, Sequence Generation</p> <p>5.7 Ethernet :wired and wireless</p>	15
6	<p><b>Network Layer</b></p> <p>6.1 IPv4 addresses: address space, Classful addressing, Classless addressing, NAT</p> <p>6.2 IPv6 addresses: Structure, address space</p> <p>6.3 IPv4: Datagram, Fragmentation, checksum, options</p> <p>6.4 IPv6: advantages, packet format, Extension headers</p>	10

**Reference Books:**

1. Computer Networks-Andrew Tanenbaum, Pearson Education [4th Edition]
2. Data Communication and Networking –Behrouz Forouzan, TATA McGraw Hill [4<sup>th</sup>Edition]
3. Networking All In One Dummies Wiley Publication [5<sup>th</sup>Edition]



Course Code: BCA-305

Course Title: Lab Course I (Data Structure)

Total Contact Hours: 48 hrs.  
(60 Lectures)

Total Credits: 04 Total Marks: 100

Teaching Scheme: Theory-05 Lectures/ Week

Assignment	Title of the Experiment	Remark
1	Non-Recursive Sorting Techniques 1. Bubble Sort 2. Insertion Sort	
2	Recursive Sorting Techniques 1. Quick Sort 2. Merge Sort	
3	Searching Techniques 1. Linear search 2. Binary Search	
4	Linked List 1. Singly Linked List 2. Singly Circular Linked List 3. Doubly Linked List 4. Doubly Circular Linked List 5. Static Linked List (Using Array)	
5	Stack 1. Static Stack Implementation 2. Dynamic Stack Implementation	
6	Queue 1. Static Queue Implementation 2. Dynamic Queue Implementation	
7	Binary Search tree (Dynamic)	
8	Graph 1. Adjacency Matrix Representation 2. Adjacency List Representation	

**Remarks and their meaning:**

- 0: Not done [ ] -Assignment is not completed by the student.
- 1: Incomplete [ ] – All exercises are not completed by the student.
- 2: Late Complete [ ] – All exercises are completed but after deadline.
- 3: Needs Improvement [ ] – Assignments are completed but the concepts are not clear.
- 4: Complete [ ] – All compulsory exercises are completed, but not case study.
- 5: Well done [ ] – All compulsory exercises are completed including CASE STUDY

## Lab Course I

Based on course contents of Data Structure using C programming is expected. To conduct practical's following is the list of sample assignments. Note that, these are only the sample assignments. Teacher may conduct practicals by preparing similar type of examples.

### Sample Assignments for LAB I

#### Assignment 1:

##### Non-recursive Sorting Techniques

1. Write a C program to accept n integers from user and sort them using bubble sort in ascending order. (Improve bubble sort algorithm is expected). Also print time complexity.
2. Write a C program to accept n integers from user and sort them using insertion sort in ascending order. Print time complexity.

##### Case Study

1. Banks often record transactions on an account, in order of the times of the transactions, but many people like to receive their bank statements with cheques listed in order by cheque number. People usually write(use) cheques in order by cheque number, and merchants usually cash them with reasonable dispatch. Thus few cheque numbers are usually out of order. Use an appropriate sorting algorithm for converting time of transaction ordering to cheque number ordering. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

##### Assignment Evaluation

0:NotDone[ ]

1:Incomplete [ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

#### Assignment 2:

##### Recursive Sorting Techniques

1. Write a C program to accept n integers from user and sort them using Quick Sort in ascending order. Print time complexity.
2. Write a C program to accept n integers from user and sort them using Merge Sort in ascending order. Print time complexity.

##### Case Study

1. Write a program to choose any random text file from within your computer and sort all words present in the file in ascending order using any of the above sorting methods.

##### Assignment Evaluation

0:NotDone[ ]

1:Incomplete [ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

#### Assignment 3:

##### Searching techniques

1. Write a C program to accept n integers from user and search any integer using linear search algorithm. Count the number of comparisons and display.

2. Write a C program to accept n integers from user and sort them using any algorithm of your choice and then modify linear search algorithm for sorted data. Count the number of comparisons.  
Compare both versions of linear search algorithm and write your findings in assignment sheet.
3. Write a C program to accept n integers from user and sort them using any algorithm of your choice and use binary search to find any integer. Print number of comparisons.

### **Case Study**

1. A spell checker is a program that looks at a document and compares each word in the document towards stored in a dictionary. If it finds words in the dictionary, it moves on to the next word, If it does not find the word, it reports the user about the misspelled (possibly) word. Formulate the problem and write a C Program to solve the problem by using appropriate data structures and algorithms.

### **Assignment Evaluation**

0:NotDone[ ]

1:Incomplete[ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

### **Assignment 4:**

#### **Linked List**

1. Write a menu driven C program to implement singly linked list of integers with following operations:
  1. Append
  2. Insert
  3. Delete
  4. Search
  5. Display
2. Write a menu driven C program to implement singly circular linked list of integers with following operations:
  1. Append
  2. Insert
  3. Delete
  4. Search
  5. Display
3. Write a menu driven C program to implement doubly linked list of integers with following operations:
  1. Append
  2. Insert
  3. Delete
  4. Search
  5. Display

4. Write a menu driven C program to implement doubly circular linked list of integers with following operations:
  1. Append
  2. Insert
  3. Delete
  4. Search
  5. Display
5. Write a menu driven program to implement static linear linked list using array. Provide all menu options as mentioned above.

### Case Study

1. A file manager identifies each file using an inode-no. A directory is also a file identified by an inode-no. Directory is a list of file search represented by an inode-no . The list has the header containing the inode –no of the parent that is the directory itself .A separate table indexed by inode-no maintains attributes of files including the name of the file. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. The union operation of two disjoint sets takes two disjoint sets S1 and S2 and returns ad joint sets consisting of all the elements of S1and S2 and the original sets S1and S2 are destroyed by the union operation. How to implement union in O(1) time using a suitable list data structure for representing a set?

OR

3. Linked list can be used to store single variable polynomial of any number of terms. Write a program to accept two polynomials with different terms to calculate their sum and multiplication.

### Assignment Evaluation

0:NotDone[ ]	1:Incomplete [ ]	2:LateComplete[ ]
3:NeedsImprovement[ ]	4:Complete[ ]	5:WellDone[ ]

### Assignment 5:

#### Stack

1. Write a C program to implement static stack of integers with following operations: Push(),Pop(), Empty(), StackTop()/Peek()
2. Write a C program to implement dynamic stack of integers with the same operations as mentioned above.

### Case Study

1. A post fix expression of the form  $ab+cd-*ab/$  is to be evaluated after accepting the values of a,b, c and d. The values should be accepted only once and the same value is to be used for repeated occurrence of same symbol in the expression. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. An Infix expression of the form  $a*(b+c)*((d-a)/b)$  need to be converted to postfix form using usual precedence's of operators. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

### Assignment Evaluation

0:NotDone[ ]

1:Incomplete [ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

### Assignment 6:

#### Queue

1. Write a C program to implement static queue of integers with following operations: Push(),Pop(), Empty(), StackTop()/Peek()
2. Write a C program to implement dynamic queue of integers with the same operations as mentioned above.

### Case Study

1. Suppose that we are selling the services of a machine. Each user pays a fixed amount peruse. However the time needed by each user is different. We wish to maximize the returns from this machine under the assumption that the machine is not to be kept idle unless no user is available. Whenever the machine becomes available, the user with the smallest time requirement is selected. When a new user requests the machine, he has to wait if there are pending requests. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

OR

2. A factory has many machines and many jobs that require processing on some of the machines. Each job has a job card which contains all the attributes of a job including the machines on which it need to be processed and the processing time. For simplicity let us assume that there are ten machines numbered 0,1,2...9, each job is processed for fixed amount of time and the job card number is coded in such away that the machine requirements can be accessed. For example if job card number is 5438,it means job requires processing on machines 5,4,3 and 8.The job can be processed simultaneously on these machines. However each machine can process jobs one at a time and takes up the new job when it becomes idle. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

### Assignment Evaluation

0:NotDone[ ]

1:Incomplete [ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

### **Assignment 7**

#### **Binary Tree**

1. Write a C program to implement binary search tree of integers with following operations:
  1. Inserting a new element
  2. Searching an element
  3. Inorder traversal
  4. Preorder traversal
  5. Postorder traversal
  6. Level-by-level traversal
  7. Counting leaf nodes
  8. Counting non-leaf nodes
  9. Creating mirror of the tree
  10. Checking whether two trees are mirrors

#### **Case Study**

1. Consider the database of books maintained in a library system. When a user wants to check whether a particular book is available, a search operation is called for. If the book is available and is issued to the user, a delete operation can be performed to remove this book from the set of available books. When the user returns the book, it can be inserted back into the set of available books. It is essential that we are able to support the above mentioned operations as efficiently as possible as since these operations are performed quite frequently. Formulate the problem and write a C program to solve the problem by using appropriate data structures and algorithms.

#### **Assignment Evaluation**

0:NotDone[ ]

1:Incomplete [ ]

2:LateComplete[ ]

3:NeedsImprovement[ ]

4:Complete[ ]

5:WellDone[ ]

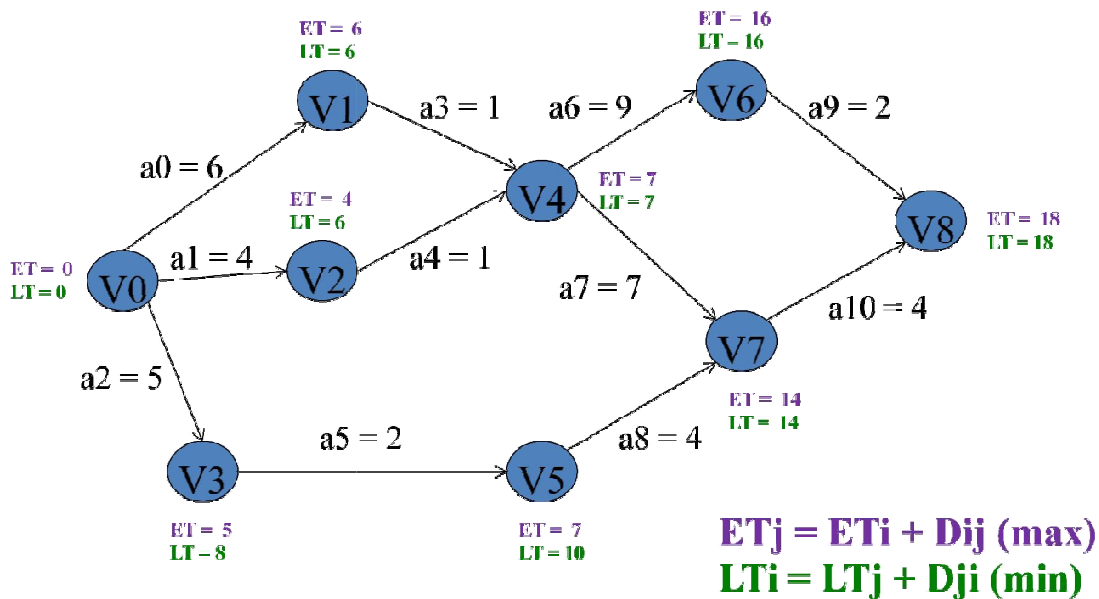
## Assignment 8:

### Graph

1. Write C program that accepts the vertices and edges for a graph and stores it as an adjacency matrix. Implement function to print In-Degree, Out-Degree and to display that adjacency matrix.
2. Write a C program that accepts the vertices and edges for a graph and stores it as an adjacency list. Implement function to print Out-Degree of any vertex  $i$ .
3. Write a C program that accepts the graph as an adjacency matrix and converts it to adjacency list representation. Write a function to display the graph in adjacency list form.

### Case Study

An activity on edge network is a directed network in which each edge models an activity with the weight of the edge representing the time needed to complete the activity. The vertex models significant events. An event represented by a vertex only happens when all of the activities modeled by edge leading into it have been completed. No activity can start until the event modeled by the vertex at its tail has occurred. The events modeled by the vertices are often project milestones such as “specification accepted by user”. Normally, we include a start vertex with in-degree 0 to model the event “Project Begins”.



Write a program to find critical path for above network.

### Assignment Evaluation

0: Not Done [ ]

1: Incomplete [ ]

2: Late Complete [ ]

3: Needs Improvement [ ]

4: Complete [ ]

5: Well Done [ ]

**Course Code: BCA-306**  
**Total Contact Hours: 48 hrs.**  
**(60 Lectures)**

**Course Title: Lab Course II**  
**Total Credits: 04 Total Marks:100**

**To conduct practicals following is the list of sample assignments. Note that, these are only the sample assignments .Teacher may conduct practical's by preparing similar type of examples**

### **Sample Assignments**

**Consider the following databases:**

#### **Bank database**

Consider the following database maintained by a Bank. The Bank maintains information about its branches, customers and their loan applications.

Following are the tables:

BRANCH (BID INTEGER, BRNAME CHAR (30), BRCITY CHAR (10))

CUSTOMER (CNO INTEGER, CNAME CHAR (20), CADDR CHAR (35), CITY (20))

LOAN\_APPLICATION (LNO INTEGER, LAMTREQUIRED MONEY,  
LAMTAPPROVED MONEY, L\_DATE DATE)

The relationship is as follows:

BRANCH, CUSTOMER, LOAN\_APPLICATION are related with ternary relationship. TERNARY (BID INTEGER, CNO INTEGER, LNO INTEGER).

#### **Student- Teacher database**

Consider the following database maintained by a college. It gives information about students and the teachers along with the subject taught by the teacher and the marks obtained by the student in the subject.

Following are the tables:

STUDENT (SNO INTEGER, S\_NAME CHAR(30), S\_CLASS CHAR(10),  
S\_ADDR CHAR(50))

TEACHER (TNO INTEGER, T\_NAME CHAR (20), QUALIFICATION  
CHAR (15), EXPERIENCE INTEGER)

The relationship is as follows:

STUDENT-TEACHER: M-M with descriptive attribute as subject name and marks.

#### **Person–Area Database**

Consider the relation Person (pnumber, pname, birthdate, income), Area (aname, area\_type). An area can have one or more persons living in it, but a person belongs to exactly one area. The attribute 'area\_type' can have values either 'urban' or



'rural'. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

Insert sufficient number of appropriate records.

### **Movie Database**

Movies(M\_name, release\_year, budget)

Actor(A\_name, role, charges, \_address)

Producer(producer\_id, name, P\_address)

Each actor has acted in one or more movies. Each producer has produced many movies and each movie can be produced by more than one producers. Each movie has one or more actors acting in it, in different roles.

Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

Insert sufficient number of appropriate records.

### **Project-Employee database**

Consider the database maintained by a company which stores the details of the projects assigned to the employees.

Following are the tables:

PROJECT (PNO INTEGER, P\_NAME CHAR(30), PTYPE

CHAR(20), DURATION INTEGER)

EMPLOYEE (ENO INTEGER, E\_NAME CHAR(20), QUALIFICATION CHAR(15), JOINDATE DATE)

The relationship is as follows:

PROJECT-EMPLOYEE: M-M Relationship, with descriptive attributes as start\_date (date), no\_of\_hours\_worked (integer).

**Business trip database** that keeps track of the business trips of salesman in an office. Salesman (sno integer, s\_name char (30), start\_year integer, deptno Varchar(10))

Trip(tno integer, from\_citychar (20), to\_citychar (20), departure\_date date, return\_date date)

Dept(deptnovarchar (10), dept\_name char(20)) Expense(eid integer, amount money)

The relation between Dept-Salesman is One to Many

The relation between Salesman-Trip : 1 to M

The relation between Trip-Expense : 1 to 1

### **Bus transport Database.**

Bus (bus\_no int , capacity int , depot\_name varchar(20))

Route(route\_no int, source char(20), destination char(20), No\_of\_stations int)

Driver(driver\_no int, driver\_name char(20), license\_no int, address Char(20), d\_age int, salary float)

Bus\_Route : M-1 and Bus\_Driver : M-M with descriptive attributes date of duty allotted and shift can be 1 (Morning) or 2 ( Evening).

Constraints: 1. License\_no is unique. 2. Bus capacity is not null.

Student –Marks database

Student(rollno integer, name varchar(30), address varchar(50), class varchar(10))

Subject (scode varchar(10), subject\_name varchar(20))

Student-Subject are related with M-M relationship with attributes marks\_scored.

### **Data Warehouse database.**

CITIES(CITY CHAR(20),STATE CHAR(20))

WAREHOUSES(WID INTEGER,WNAME CHAR(30),LOCATION CHAR(20))

STORES(SID INTEGER,STORE\_NAME CHAR(20), LOCATION\_CITY CHAR(20))

ITEMS(ITEMNO INTEGER,DESCRIPTION TEXT,WEIGHT DECIMAL(5,2), COST DECIMAL(5,2) )

CUSTOMER(CNO INTEGER, CNAME CHAR(50),ADDR VARCHAR(50), CU\_CITY CHAR(20))

ORDERS(ONO INT, ODATE DATE)

The relationship is as follows

CITIES-WAREHOUSES 1 TO M

WAREHOUSES-STORES 1 TO M

CUSTOMER– ORDERS 1 TO M

ITEMS– ORDERS M TO M relationship with descriptive attribute ordered\_quantity

STORES-ITEMS M TO M RELATION with descriptive attribute quantity

## Assignment 1: Bank Database

### Bank database

Consider the following database maintained by a Bank. The Bank maintains information about its branches, customers and their loan applications.

Following are the tables:

BRANCH (BID INTEGER, BRNAME CHAR (30), BRCITY CHAR (10))

CUSTOMER (CNO INTEGER, CNAME CHAR (20), CADDR CHAR (35), CITY (20))

LOAN\_APPLICATION (LNO INTEGER, LAMTREQUIRED MONEY, LAMTAPPROVED MONEY, L\_DATE DATE)

The relationship is as follows:

BRANCH, CUSTOMER, LOAN\_APPLICATION are related with ternary relationship.

TERNARY (BID INTEGER, CNO INTEGER, LNO INTEGER).

### Nested Queries

Solve the Queries

1. Find the names of the customers for the "M.G.ROAD" branch.
2. List the names of the customers who have received loan less than their requirement.
3. Find the maximum loan amount approved.
4. Find out the total loan amount sanctioned by "Camp" branch.
5. Count the number of loan applications received by "M.G.ROAD" branch.
6. List the names of the customer along with the branch names who have applied for loan in the month of September.

### Views:

1. Create a view which contains the details of all customers who have applied for a loan more than Rs.500000.
2. Create a view which contains details of all loan applications from the 'Shivajinagar' branch.
3. Write the following Queries, on the above created views :
  - a. List the details of customers who have applied for a loan of Rs. 500000.
  - b. List the details of loan applications from 'Shivajinagar' , where loan amount is >Rs 50000.

c. List the details of Loan applications, with the same loan amount.

**Stored Functions:**

1. Write a function that returns the total number of customers of a particular branch.

(Accept branch name as input parameter.)

2. Write a function to find the maximum loan amount approved.

**Errors and Exceptions:**

1. Write a stored function to print the total number of customers of a particular branch.

(Accept branch name as input parameter.) In case the branch name is invalid, raise an exception for the same.

2. Write a stored function to increase the loan approved amount for all loans by 20%. In case the initial loan approved amount was less than Rs 10000, then print a notice to the user, before updating the amount.

**Triggers**

1. Write a trigger before deleting a customer record from the customer table. Raise a notice and display the message “customer record is being deleted”

2. Write a trigger to ensure that the loan amount entered for a customer of ‘Shivajinagar’ branch is never < 1000 and greater than 1000000.

## Assignment 2: Student-Teacher Database

### Student- Teacher database

Consider the following database maintained by a college. It gives the information about students and the teachers along with the subject taught by the teacher and the marks obtained by the student in the subject.

Following are the tables:

STUDENT (SNO INTEGER, S\_NAME CHAR(30), S\_CLASS CHAR(10), S\_ADDR CHAR(50))

TEACHER (TNO INTEGER, T\_NAME CHAR (20), QUALIFICATION CHAR (15), EXPERIENCE INTEGER)

The relationship is as follows:

STUDENT-TEACHER: M-M with descriptive attribute as subject name and marks.

### Nested Queries

Solve the queries

1. Find the minimum experienced teacher.
2. Find the number of teachers having qualification "Ph. D."
3. List the names of the students taught by "Mr. Kumar" along with the subjects taught.
4. Find the subjects taught by each teacher.
5. List the names of the teachers who are teaching to a student named "Suresh".
6. List the names of all teachers along with the total number of students they are teaching.
7. Find the student having maximum marks in the subjects taught by "Mr. Kumar".

### Views

1. Create a view containing details of all the teachers teaching the subject 'Mathematics'.
2. Create a view to list the details of all the students who are taught by a teacher having experience of more than 3 years .
3. Write the following Queries, on the above created views :
  - a. List the name of the most experienced teacher for "Mathematics".
  - b. List the names of students of 'S.Y.B.C.A.' class, who are taught by a teacher having more than 3 years experience.

### **Stored Functions**

1. Write a function to accept teacher name as input and returns the number of students taught by the teacher.
2. Write a function to accept name of subject and count the number of teachers who teach that subject.
3. Write a function to accept student name and calculate the total marks obtained by that student.

### **Cursors**

1. Write a stored function using cursors to accept student name from the user and find the names of all teachers and subjects taught to the student.
2. Write a stored function using cursors which will calculate total number of subjects taught by each teacher.

### **Errors and Exceptions**

1. Write a stored function to accept teacher name as input parameter and print the total number of students taught by the teacher. In case the teacher name is invalid, raise an exception for the same.
2. Write a stored function to increase the marks of each student to 40 if the marks are between 35 and 40. Print a notice to the user, before updating the marks.

### **Triggers**

1. Write a trigger before deleting a student record from the student table. Raise a notice and display the message "student record is being deleted"
2. Write a trigger to ensure that the marks entered for a student, with respect to a subject is never  $< 0$  and greater than 100.

### Assignment 3: Movie Database

Q1. Consider the following Movie database.

Movies (m\_name varchar (25), release\_year integer, budget money)

Actor (a\_name char (30), role char (30), charges money, a\_address varchar (30) )

Producer(producer\_id integer, name char (30), p\_address varchar (30) )

Each actor has acted in one or more movies. Each producer has produced many movies and each movie can be produced by more than one producers. Each movie has one or more actors acting in it, in different roles.

Create the above database in PostGreSQL.

#### Execute the following queries in PostGreSQL

- 1) List the names of actors who have acted in at least one movie, in which Mr. Khan has acted.
- 2) List the names of the actors and their movie names.
- 3) List the names of movies whose producer is "Mr. Khan"
- 4) List the names of the movies with the highest budget.
- 5) List the names of movies released after 2000.
- 6) List the names of actors who played the role of 'Villan'.
- 7) List the names of actors who are given the maximum charges for their movie along with movie name & release year.
- 8) Display count & total budget of all movies released in year 2014.

#### Cursor and triggers

1. Write a trigger before inserting into a movie table to check budget. Budget should be minimum 50 lakh. Display appropriate message.
2. Write a stored function using cursors to display the names of actors who have acted in the maximum number of movies.

#### Stored functions

1. Write a function to list movie-wise charges of 'Amitabh Bachchan'
2. Write a stored function to accept producer name as input and print the names of movies produced by him/her.. Also print the total number of actors in that movie.
3. Write a stored function to accept movie name as input and print the names of actors working in the movie.

#### View

- 1) List the names of producers who produce the same movie as 'Mr. Karan Johar' has produced.
- 2) List the names of actors who do not live in Mumbai or Pune city.
- 3) List the actors in each movie.

## **Exception**

1. Write a stored function to accept movie name as input and print the names of actors working in the movie. Also print the total number of actors working in that movie. Raise an exception for an invalid movie name.
2. Write a stored function to accept producer name as input and print the names of Movies produced by him/her. Also print the total number of actors in that movie. Raise an exception for an invalid producer name.



### Assignment 4: Business Trip Database

Q1. Consider the following Business trip database that keeps track of the business trips of salesman in an office.

Salesman (sno integer, s\_name char (30), start\_year integer, deptno Varchar(10))

Trip(tno integer, from\_city char (20), to\_city char (20), departure\_date date, return\_date date)

Dept (deptno varchar (10), dept\_name char(20))

Expense(eid integer, amount money)

The relation between Dept-Salesman is One to Many

The relation between Salesman-Trip : 1 to M

The relation between Trip-Expense : 1 to 1

Create the above database in PostgreSQL.

#### Query

##### Execute the following queries in PostgreSQL

1. Give the details for trips that exceed Rs. 10,000 in expenses.
2. Increase the expenses of all the trips by Rs. 5000.
3. Give the details of salesmen who started working before 2000.
4. List the salesmen whose names start with 'A'.
5. List the salesman numbers and names of the salesmen who made trips to Calcutta.
6. Find the departments from which the salesmen have done highest number of trips.
7. Find the total expenses incurred by the salesman "Mr. Patil".
8. List the names departments, that have salesman, who have done maximum numbers of trips.

#### Cursor and trigger

1. Write a cursor to list all salesman's name, his trip details & expenses.
2. Write a stored function using cursors to display the details of a salesman information. (Accept salesman name as input parameter).
3. Write a trigger before inserting into a trip table to check departure date. 'departure date' should be always > current date. Display appropriate message.
4. Delete all the trips made by department "computer" having expenses more than Rs.15000

#### Stored Procedure

1. Write a function to find a business trip having maximum expenses.
2. Write a function to count the total number of business trips from 'Pune' to 'Mumbai'.

3. Write a stored function to accept department name as input and print the names of salesmen working in the department. Also print the total number of trips of those salesmen.
4. Write a function to list the salesman numbers and names who made trips to Calcutta.

#### **view**

1. List the names of departments that have salesmen, who have done maximum number of trips.
2. Find the total expenses incurred by the salesman “\_”.
3. Create a view to list the details of all salesman from ‘Western’ department.
4. Create a view to list all salesman’s name his trip details and his expenses for the trip.
5. Write the following Queries, on the above created views :
6. List the details of salesmen from ‘Western’ department, whose start year is 2005.
7. List the names of salesmen from ‘Western’ department, for whom their total trip expense is >Rs 100000.
8. List the names of salesmen who have gone on a trip to “Mumbai” city.

#### **Exception**

1. Write a stored function to accept department name as input and print the names of salesmen working in the department. Also print the total number of trips of those salesmen. Raise an exception for an invalid department name.
2. Write a stored function to accept the trip\_no and departure date and print its expenses. Raise an exception in case of invalid trip number.

## Assignment 5: Bus Transport Database

Q1. Consider the following Bus transport Database.

Bus (bus\_no int, capacity int, depot\_name var char(20))

Route (route\_no int, source char(20), destination char(20), No\_of\_stations int)

Driver (driver\_no int, driver\_name char(20), license\_no int, address Char(20), d\_age int, salary float)

Bus\_Route : M-1 and Bus\_Driver : M-M with descriptive attributes date of duty allotted and shift can be 1 (Morning) or 2 (Evening).

Constraints: 1. License\_no is unique. 2. Bus capacity is not null.

Create the above database in PostgreSQL.

### Query

#### Execute the following queries in PostgreSQL

1. Find out the name of the driver having maximum salary.
2. Delete the record of bus having capacity < 10
3. Increase the salary of all drivers by 5% if driver's age > 45.
4. Find out the route details on which buses of capacity 20 run.
5. Print the names & license nos. of drivers working on in both shifts.
6. Print the names of drivers working in both shifts on .

### Cursor and Trigger:

1. Define a trigger after insert or update the record of driver if the age is between 18 and 50 give the message "valid entry" otherwise give appropriate message.
2. Define a trigger after delete the record of bus having capacity < 10. Display the message accordingly.
3. Define a trigger after insert or update the record of driver if the age is between 18 & 50 give the message "valid entry" otherwise give appropriate message.
4. Write a stored function using cursors to display the details of a driver. (Accept driver name as input parameter).

### Stored function

1. Write a stored function to print the names of drivers working on both shifts on '20/04/2014'.
2. Write a stored function to display the details of a driver. (Accept driver name as input parameter).
3. Write a function to accept the bus\_no, date & print its allotted driver.

**View**

1. which contains details of bus no 101 along with details of all drivers who have driven that bus.
2. To display the details of the buses that run on routes 1 or 2.
3. To find out the name of the driver having maximum salary.
4. To accept the bus\_no and date and print its allotted driver.

**Exception**

1. Define a trigger after insert or update the record of driver if the age is between 18 and 45 give the message "valid entry" otherwise raise exception.
2. Write a stored function to accept the bus\_no and date and print its allotted drivers. Raise an exception in case of invalid bus number.

## Assignment 6: Student-Marks Database

Q1. Consider the following Student –Marks database

Student(rollno integer, name varchar(30), address varchar(50), class varchar(10))

Subject (scode varchar(10), subject\_name varchar(20))

Student-Subject are related with M-M relationship with attributes marks\_scored.

Create the above database in PostgreSQL.

### Query

Execute the following queries in PostgreSQL

1. Display the names of students scoring the maximum total marks.
2. List the distinct names of all the subjects.
3. Display class wise & subject wise student list.

### Cursor and triggers

1. Write a stored function using cursors, to accept a address from the user and display the name, subject and the marks of the students staying at that address.
2. Write a stored function using cursors which will calculate total and percentage of each student.
3. Write a trigger before deleting a student record from the student table. Raise a notice and display the message “student record is being deleted”
4. Write a trigger to ensure that the marks entered for a student, with respect to a subject is never < 10 and greater than 100.

### Stored function

1. Write a stored function using cursors to accept an address from the user & display the name, subject & the marks of the student staying at that address.

### View

1. To list student name, class & total marks scored by each student ,sorted by student name.
2. To list student names along with subject name and marks who scored more than 40 marks.
3. Containing all the details of student named ‘Omkar’.

## Assignment 7: Warehouse Database

Q1. Consider the Warehouse database.

CITIES(CITY CHAR(20),STATE CHAR(20))

WAREHOUSES(WID INTEGER,WNAME CHAR(30),LOCATION CHAR(20))

STORES(SID INTEGER,STORE\_NAME CHAR(20), LOCATION\_CITY CHAR(20))

ITEMS(ITEMNO INTEGER,DESCRIPTION TEXT,WEIGHT DECIMAL(5,2), COST DECIMAL(5,2) )

CUSTOMER(CNO INTEGER, CNAME CHAR(50),ADDR VARCHAR(50), CU\_CITY CHAR(20))

ORDERS(ONO INT,ODATE DATE) The relationship is as follows

CITIES-WAREHOUSES 1 TO M

WAREHOUSES-STORES 1 TO M

CUSTOMER- ORDERS 1 TO M

ITEMS- ORDERS M TO M relationship with descriptive attribute ordered\_quantity

STORES-ITEMS M TOM RELATION with descriptive attribute quantity

### Query

**Solve the following queries.**

1. Find the item that has minimum weight.
2. Find the different warehouses in "Pune".
3. Find the details of items ordered by a customer "Mr. Patil".
4. Find a Warehouse which has maximum stores.
5. Find an item which is ordered for minimum number of times.
6. Find the details orders given by each customer.

### Cursors and triggers:

1. Write a stored function using cursors to accept a city from the user and to list all warehouses in the city.
2. Write a stored function using cursors to find the list of items whose cost is between Rs.5000 to 10000

### Views:

1. Create a view containing details of all the stores of a Warehouse named 'Spares'.
2. Create a view to list the details of all customers who have placed orders on the date '03-10-2013' .
3. 'List the names of stores of 'Spares ' warehouse, located at Pune.
4. List the names of customers from Pune city, who have placed orders on 03-10-2013
5. List the orders placed by "Mr. Joshi".

## Course Title: Practical Assignment for 304: Computer Networking

### I. Execute the following commands and write their output

#### 1. ping

This command is used to test connectivity between two nodes. Ping is use to communicate to other devices. You can ping host name or IP address using the following commands.

example: ping 195.55.100.1 or ping www.google.com

```
$ping <server-ip-address>
```

Output:

```
$ping localhost
```

Output:

```
$ping <other-ip-in-network>
```

Output:

#### 2. hostname

Gives the host name of the computer they are logged into. To set the hostname permanently use /etc/sysconfig/network file.

```
$hostname
```

Output :

hostname with no options displays the machine's hostname

hostname -d displays the domain name the machine belongs to

hostname -f displays the fully qualified host and domain name

hostname -i displays the IP address for the current machine

#### 3. traceroute

traceroute is a network troubleshooting utility which shows number of hops taken to reach destination also determine packets traveling path.

```
$traceroute ip-address
```

Output :

#### 4. netstat



Netstat (Network Statistic) command displays interfaces, connection information, routing table information etc.

```
$netstat
```

Output :

Execute it with the following options and write the output:

```
netstat -t
```

```
netstat -s -t
```

```
netstat -i
```

## 5. ifconfig

ifconfig is used for displaying network interface information. ifconfig (interface configurator) command is use to initialize an interface, assign IP Address to interface and enable or disable interface on demand. With this command you can view IP Address and Hardware / MAC address assign to interface and also MTU (Maximum transmission unit) size.

```
$/sbin/ifconfig
```

Output :

## 6. who

Displays information of all users who are logged in.

```
$who
```

Output :

## 7. whoami

The whoami command writes the user name (i.e., login name) of the owner of the current login session to standard output.

```
$whoami
```

Output :

## 8. nmap

Network mapper tool to discover hosts and services on a computer network.

```
$ nmap <ip-address>
```

Output :

```
$nmap <server-ip-address>
```

Output :

## 9. nslookup

If you know the IP address it will display hostname. To find all the IP addresses for a given domain

name, the command nslookup is used. You must have a connection to the internet for this utility to be useful, e.g.

\$nslookup [www.google.com](http://www.google.com)

## 10. Route

**route** command also shows and manipulate **ip** routing table. To see default routing table in **Linux**, type the following command.

\$route

Output:

## Study of LAN environment

Find out information about the network in your lab and fill in details below:

1. Total Number of computers in your lab:
2. Find details of any 5 computers :

	MAC address	IP address	hostname	LAN speed
1				
2				
3				
4				
5				

3. Are the IP addresses assigned to the machines statically or dynamically?
4. Does the network have a DHCP server?
5. If yes, what is the address of the server ?
6. How many servers are configured? :

Details of servers : IP address, MAC address, Purpose

	IP address	MAC address	Purpose
1			
2			

## 7. Cables

- a. Type :
- b. Is it coaxial / twisted pair or fiber optic cable ?
- c. Cable bandwidth
- d. Maximum cable length limit
- e. Connector used

## 8. Switches:

Write following information about switches:  
Company, MAC address, Number of port connected.

9. Is there wi-fi capability in the LAN?

If yes,

- i. What is the Wi-fi access point address?
- ii. How many devices / IP addresses does it support?
- iii. What is the bandwidth?

If no,

- iv. What additional devices are needed?
- v. Where will you connect them?
- vi. What will be its IP address?

10. Is there internet access in the lab?

If not, what changes to the hardware / software must be made ?

If yes, what is the IP address of the router / gateway ?

11. Draw the Network Topology (show how machines and servers are connected using connectivity devices)

12. If 10 more machines have to be added to the network, what changes must be made to the network?

13. If the network is to be divided into four subnetworks having 50 machines each, give a plan to do so. What additional devices will be needed ? Give the IP address of each subnetwork and the address ranges for hosts in each subnetwork.

**Savitribai Phule Pune University**  
**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

Course Code: BCA 401  
Total Contact Hours: 60hrs  
Total Marks: 100

Course Title: C++  
Total Credits: 04  
Teaching Scheme: Theory- 05 Lect. /Week

**Course Objectives:**

- Understand object oriented programming:
  - Be able to explain the difference between object oriented programming and procedural programming.
  - Be able to program using C++ features such as Class, objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
  - Be able to build C++ classes using appropriate encapsulation and design principles.
- Improve problem solving skills:
  - Be able to apply object oriented or non-object oriented techniques to solve bigger computing problems

**Prerequisites:** Knowledge of C Programming Language

<b>Unit No</b>	<b>Contents</b>	<b>No Of Lectur</b>
1.	<b>Introduction to C++</b> 1.1. Basics of C++, 1.2. Structure of C++ Program, keywords in C++, 1.3. Data types hierarchy in C++, 1.4. Operators in C++: Scope resolution operator, Insertion and Extraction operator New and Delete operators. 1.5. Reference variable. 1.6. Manipulators function: endl, setw, set fill, set precision.	[6]
2.	<b>Object oriented Concepts</b> 2.1. Object oriented concepts 2.2. Features, 2.3. Advantages and Applications of OOP 2.4. Difference between Procedure oriented programming and object oriented programming.	[2]

3.	<b>Classes and Objects</b> 3.1. Structure sand class, Class, Object, Access specifies, 3.2. Class members, 3.3. Defining member functions :Inside and outside the class definition, 3.4. Creating objects. String class, operation on string, Array of objects. 3.5. 'this' pointer.	[10]
4.	<b>Function in C++</b> 4.1. Call by reference, Return by reference, 4.2. Function overloading and default arguments 4.3. Inline function 4.4. Passing and returning objects from functions, Static class members 4.5. Friend Concept – Function, Class	[6]
5.	<b>Constructors and Destructors</b> 5.1. Memory allocation and static data members 5.2. Definition of constructor Types of constructors: Default Constructor 5.3. Constructor with default arguments 5.4. Parameterized Constructor 5.5. Copy Constructor 5.6. Overloaded constructors in a class 5.7. Destructors	[4]
6.	<b>Operator overloading</b> 6.1. Introduction, rules of operator overloading 6.2. Operator overloading: 6.3. Unary and binary operators, 6.4. Comparison, arithmetic assignment operator 6.5. Overloading new & delete operators 6.6. Overloading without friend function and using friend function,	[6]
7.	<b>Inheritance</b> 7.1. Introduction 7.2. Types of Inheritance: Single inheritance Multiple inheritance, Multilevel inheritance Hierarchical inheritance Hybrid inheritance. 7.3. Derived Class Constructor sand Destructors 7.4. Ambiguity in multiple Inheritances, virtual base classes, Abstract base class.	[8]

8.	<b>Virtual Function &amp; Polymorphism</b> 8.1. Introduction, Pointer to object, Pointer to derived 8.2. class, Overriding member functions, Virtual function, Rules for virtual functions, pure virtual function, Run-time type information (RTTI)	[6]
9.	<b>Working with files</b> 9.1. File operations – Text files, Binary files 9.2. File stream class and methods 9.3. File updation with random access 9.4. Overloading insertion and extraction operator	[4]
10.	<b>Templates</b> 10.1 Introduction function templates, function templates with multiple parameters 10.2 Overloading of template functions, Class Templates, class template multiple Parameters, member function templates 10.3 Introduction to Standard Template Library (STL) 10.4 Components of STL 10.5 Containers 10.6 Algorithms 10.7 Iterators 10.8 Application of Container classes	[5]
11.	<b>Exception handling</b> 11.1 Exception Handling Mechanism 11.2 The try block 11.3 The catch block 11.4 Throw statement 11.5 The try/throw/catch sequence	[3]

### Reference Books:-

1. Object Oriented Programming with C++ by Robert Lafore
2. Object Oriented Programming with C++ by E. Balagurusamy
3. The Complete Reference C++ by Herbert Schildt
4. Let us C++ by– Yashwant Kanitkar
5. Mastering C++ by Venugopal, TRavishankar, Rajkumar THM Pub.
6. Thinking in C++ 2nd Edition by Bruce Eckel, Prentice Hall Pub.

**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

Course Code: BCA 402  
Total Contact Hours: 48hrs.  
Total Marks: 100

Course Title: Introduction to Web Technology  
Total Credits: 04  
Teaching Scheme: Theory- 05 Lect. /Week

<b>Unit No</b>	<b>Contents</b>	<b>No Of Lectures</b>
1	<b>Introduction to Web Technologies</b> 1.1 Introduction to Web Technologies 1.2 How the Website Works? 1.3 Clients- Servers and Communication 1.4 Client and Server Scripting Languages 1.5 Internet-Basic, Internet Protocols(HTTP,FTP,IP) 1.6 World Wide Web(WWW) 1.7 HTTP request message, HTTP response message 1.8 Types of Websites(Static and Dynamic Websites)	7
2	<b>Advance HTML tags</b> 2.1 HTML Tags. 2.2 Embedding Audio and Video. 2.3 Introduction to elements of HTML 2.4 Working with Text 2.5 Working with Lists, Tables and Frames 2.6 Working with Hyperlinks, Images and Multimedia 2.7 Working with Forms and controls 2.8 Concept of CSS, Types of CSS 2.9 Creating Style Sheet	12

3	<b>JAVA Script</b> 3.1 Introduction to client side scripting. 3.2 Introduction to Java Script. 3.3 Identifier & operator, control structure. 3.4 Using Functions. 3.5 Event Handling in Java Scripts 3.6 Document object model(DOM), 3.7 DOM Objects(window, navigator, history, location) 3.8 Predefined functions, math & string functions 3.9 Array in Java scripts 3.10 Event handling in Java script 3.11 Sufficient examples to be covered.	12
5	<b>Arrays in PHP</b> 4.1 Indexed Vs Associative arrays 4.2 Identifying elements of an array 4.3 Storing data in arrays 4.4 Multidimensional arrays 4.5 Extracting multiple values 4.6 Converting between arrays and variables 4.7 Traversing arrays 4.8 Sorting 4.9 Action on entire arrays 4.10 Using Arrays.	10
6	<b>Function in PHP</b> 5.1 Defining and calling a function 5.2 Default parameters , Variable parameters, Missing parameters Variable function, Anonymous function.	7

**Reference Books : -**

1. Complete HTML- Thomas Powell
2. HTML and JavaScript–Ivan Bayross
3. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
4. Beginning PHP5, Wrox publication
5. PHP for Beginners, SPD publication



**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

Course Code: BCA 403

Course Name: Advanced Networking and Network Security

Total Contact Hours: 48hrs.

Total Credits: 04

Total Marks: 100

Teaching Scheme: Theory- 05

Lect./Week

Unit No.	Contents	No. of Lectures
1	<p><b>Transport Layer</b></p> <p>1.1. Process-to-Process Delivery Client Server Paradigm, Multiplexing and De-multiplexing, Connectionless Vs Connection-Oriented Service, Reliable Vs Unreliable</p> <p>1.2. User Datagram Protocol(UDP) Datagram Format, Checksum, UDP operations, Use of UDP</p> <p>1.3. Transmission Control Protocol (TCP) TCP Services –Process to-Process Communication, Stream Delivery Service, sending and Receiving Buffers, Segments, Full –Duplex Communication, Connection oriented service, Reliable service</p> <p>1.4. TCP Features –Numbering System, Byte Number, Sequence Number, Acknowledgement Number, Flow Control, Error Control, Congestion Control</p> <p>1.5. TCP Segment Format</p>	8
2	<p><b>Computer Security</b></p> <p>2.1. Introduction</p> <p>2.2. Need for security</p> <p>2.3. Principles of Security</p> <p>2.4. Types of Attacks</p> <p>2.5. Cryptography</p> <p>2.6. Plaintext and Cipher Text</p> <p>2.7. Substitution techniques</p> <p>2.8. Caesar Cipher</p> <p>2.9. Mono-alphabetic Cipher</p> <p>2.10. Polygram, Poly alphabetic Substitution</p> <p>2.11. Play fair</p> <p>2.12. Hill Cipher</p> <p>2.13. Transposition techniques,</p> <p>2.14. Encryption and Decryption,</p> <p>2.15. Symmetric and Asymmetric Key Cryptography</p> <p>2.16. Steganography</p>	10

	2.17. Key Range and Key Size 2.18. Possible Types of Attacks ,Attacks on cipher text	
3	<b>Symmetric Key Algorithm</b> 3.1. Algorithms 3.2. Types and modes 3.3. Overview of Symmetric key Cryptography 3.4. Data Encryption Standard (DES),Types of DES 3.5. Diffie Helman Key Exchange Algorithm	8
4	<b>Asymmetric Key Algorithm</b> 4.1. Brief history of Asymmetric Key Cryptography 4.2. Overview of Asymmetric Key Cryptography 4.3. RSA algorithm 4.4. Symmetric and Asymmetric key cryptography 4.5. Digital Signature 4.6. Message Digest and its uses 4.7. Problems with the public key exchange	12
5	<b>Internet Security Protocols</b> 5.1 Digital Certificates 5.2 Basic concepts of Internal security 5.3 Secure Socket Layer (SSL) 5.4 Transport Layer Security (TLS) 5.5 Secure Hyper Text Transfer Protocol (SHTTP) 5.6 Time Stamping Protocol (TSP)	6
6	<b>Network Security Firewalls and Virtual Private Networks</b> 6.1. Brief Introduction to Firewalls, IP Security, Virtual Private Networks (VPN) 6.2. Intrusion 6.3. User Authentication: 6.3.1 Authentication basics 6.3.2 Passwords, Authentication, Tokens 6.3.3 Biometric Authentication 6.3.4 Image-Base Authentication	12

#### Reference Books:-

1. Cryptography and Network Security by Atul Kahate, 3<sup>rd</sup> Edition, Tata McGraw Hill
2. Cryptography and Network Security by William Stallings, Fifth Edition, Pearson Education.
3. Cryptography: Theory and Practice by Douglas Stinson, CRC Press, CRC Press LLC.

**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

Course Code: BCA-404  
Total Contact Hours: 48hrs.  
Total Marks: 100

Course Title: OOSE  
Total Credits: 04  
Teaching Scheme: Theory- 05 Lect./Week

Unit No	Content	No. Of Lectures
1	<b>Review of Object Oriented Concepts</b> 1.1 System Concepts – Software Engineering Concepts 1.2 Review of object orientation: classes and objects, polymorphism ,inheritance 1.3 classes and relationship 1.4 Interfaces.	[4]
2	<b>Introduction to Modeling and UML</b> 2.1 Importance of Modeling 2.2 Principles of Modeling, Object Oriented Modeling. 2.3 Overview of UML 2.4 Conceptual Model of UML 2.5 Architecture.	[4]
3	<b>Structural Modeling</b> 3.1 Classes and Advance Classes 3.2 Relationships and Advance Relationships 3.3 Interfaces, Types and Roles. 3.4 Packages 3.5 Common Mechanisms 3.6 Diagrams	[8]
4	<b>Structural Diagrams</b> 1 ClassDiagrams4.2Instances 4.3 Object Diagrams 4.4 Case Study(Minimum Two)	[8]

5	<b>Behavioral Modeling</b> 5.1 Interactions 5.2 Use Cases 5.3 Events and Signals 5.4 Processes and Threads 5.5 Time and Space	[4]
6	<b>Behavioral Diagrams</b> 6.1 Use Case Diagrams 6.2 Interaction Diagrams 6.3 Activity Diagrams 6.4 State machines 6.5 State chart Diagrams 6.6 Case Study(Minimum Two)	[14]
7	<b>Architectural Modeling</b> 7.1 Components 7.2 Deployment 7.3 Collaborations 7.4 Pattern and framework 7.5 Component diagram 7.6 Deployment Diagram 7.7 Case Study(Minimum Two)	[8]
8	<b>Case Study</b> Detail Case Studies for practice including ( but not limited) following areas: ATM/Library Management/Hotel Management / Book Store / Document Editor/College Administration/ Railway Reservation/Vending Machine etc.	[6]

**Reference Books:-**

- 1) The Unified Modelling Language Reference Manual-James Rambaugh Ivar Jacobson, Grady Booch
- 2) Applying UML and Patterns :An Introduction to Object Orented Analysis and Design and Iterative Development –Craig Larman

**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

Course Code: BCA-405  
Total Contact Hours: 48 hrs.  
Total Marks: 100  
Lect./Week

Course Title: LAB I (C++ Practical)  
Total Credits: 04  
Teaching Scheme: Theory- 05

**Note that these are only sample assignments. Teachers may conduct practical's by preparing similar types of examples**

**Sample Assignments for C++**

**Topic: Class, Array of Object, Dynamic Memory Allocation, Manipulators**

1. Create a class for inventory of books containing author, title, price, publisher and stock as data members. Book can be sold, if stock is available, otherwise purchase will be made. Write necessary member functions for the following:
  - a. To accept details from user
  - b. To sale a book. (Sale contains book details & number of copies to be sold.)
  - c. To Purchase a book. (Purchase contains book details & number of copies to be purchased)  
(Use new operator to allocate memory).
2. Write a C++ program to read a set of 'n' numbers (accepted from user) and print the contents of the array in the reverse order. (Use new and delete operator).
3. Write a C++ program using class called 'Point' that has x & y as integer data members. The class has the following member functions:

void setpoint(int, int)	to set the specified values of x and y in object.
Void showpoint()	to display contents of point object.
Point addpoint (point)	to add the corresponding values of x and y in Point object argument to current point object return point.
4. Write a program that consists of two classes Time12 and Time24. The first one maintains time on a 12-hour basis, whereas the other one maintains it on a 24-hour basis. Provide conversion functions to carry out the conversion from object of one type to another.
5. Define a class to represent a bank account which includes following members :
  - Data members –
    - a. Name
    - b. Account number

- c. Type of account
  - d. Bal. amt
    - Member functions –
      - a. To assign initial value
      - b. To deposit an account
      - c. To withdraw an account
      - d. To display name, account number & balance.
6. Write class sales (Salesmam\_Name, Product\_name, Sales\_Quantity, Target). Each salesman deals with a separate product and is assigned a target for a month. At the end of the month his monthly sales is compared with target and commission is calculated as follows:
- a) If Sales\_Quantity > target then commission is 30% of extra sales made + 15% of target
  - b) If Sales\_Quantity == target then commission is 20% of target.
  - c) Otherwise commission is zero
  - d) Display the sales information along with commission obtained. (Use array of objects)
7. Write a C++ program to create a class Employee which contains data members as Emp\_Id, Emp\_Name, Basic\_Salary, HRA, DA, Gross\_Salary. Write member functions to accept Employee information. Calculate and display Gross salary of an employee. (DA=15% of Basic salary and HRA = 25% of Basic salary) (Use appropriate manipulators to display employee information in given format :- Emp\_Id and Emp\_Name should be left justified and Basic\_Salary, HRA, DA, Gross salary Right justified with a precision of two digits for specified width)
8. Write a C++ program to create a class Person which contains data members as P\_Name, P\_City, P\_Contact\_Number. Write member functions to accept and display five Persons information. Design User defined Manipulator to print P\_Contact\_Number. (For Contact Number set right justification, maximum width to 10 and fill remaining spaces with '\*')
9. Write a C++ program to accept 'n' numbers from user through Command Line Argument. Store all even and odd numbers in two different arrays. Display contents of both arrays.
10. Write a C++ program to create a class Student which contains data members as Roll\_Number, Stud\_Name, Marks in five subjects. Write member functions to accept Student information. Display all details of student along with a percentage and class obtained depending on percentage. (Use array of objects)

11. Consider a class Complex

```

Class Complex
{
    float real;

    float imaginary;
Public:

```

```
//methods;
```

```
};
```

Write the member function to subtract two Complex Numbers (Object as argument).

12. Create a C++ class for a student object with the following attributes—roll no, name, number of subjects, marks of subjects. Write member function for accepting marks and display all information of student along with total and Percentage. Display marklist with Use of manipulators.
13. Write a necessary class & member function definition for a cricket player object with data members as P\_code, P\_name, runs, innings\_ played and number\_of\_times\_not\_out. (Use array of objects).

The program should contain following menu:

- Enter details of players.
- Display details and average runs of all players.

14. Create a class Time that contains hours, minute and seconds as data members. Write the member functions:
  1. to convert a time into total number of seconds.
  2. to display the time into format like: 09:30:50

15. Write a class to represent a vector. Include member functions to perform the following tasks:

- a. Create the vector
- b. Modify the value of given element.
- c. Multiply by a scalar value
- d. Display the vector in the form (10, 20, 30...)

16. Consider the following class Person

```
Class Person
```

```
{
```

```
    char Name [20];
```

```
    char Addr [30];
```

```
    float Salary;
```

```
    int Property; //in sq. foot area
```

```
    float tax_amount;
```

```
Public:
```

```
    // methods
```

```
};
```

- a) Calculate tax amount by checking salary and the property of the person
- i. For salary  $< 5000$  tax rate = 0
  - ii. For salary  $\geq 5000 \parallel \leq 10000$  tax rate = 14% of salary.
  - iii. For salary  $\geq 10000$  tax rate = 16% of salary.
- b) In this tax amount add following amt depending on the size of area in sq. foot
- i. For 1000 sq. foot area amt = 0.
  - ii. For  $> 1000 \parallel < 5000$  sq. foot area amt = 1000
  - iii. For  $> 5000 \parallel \leq 10000$  sq. foot area amt = 3000.



## **Topic:-Inline Function, Function Overloading and Friend function**

1. Write a C++ program using class with data member's int feet, float inches to represent distance and define function that takes two distance values as arguments and returns the larger one. Include a main program that accepts two distance figures from the user compare them and displays the larger using Inline function.
2. Write a menu driven C++ program using class to perform all arithmetic operation (+,-,\*,/) (use inline function).
3. Write a C++ program to find volume of cube, cylinder and rectangle using function overloading
4. Write a C++ program to find area of triangle, circle, and rectangle using function overloading.
5. Create a class student containing data members:
  - a. Roll\_no
  - b. name
  - c. marks1, marks2, marks3

Write necessary member functions:

- a. to accept details of all students
  - b. to display details of one student
  - c. to display details of all students (Use Function overloading).
6. Write a C++ program to calculate area of Circle, Sphere and cylinder using function overloading.
  7. Create a class telephone containing name, telephone number & city as data members and write necessary member functions for the following:
    - a. Search the telephone number with given name.
    - b. Search the name with given telephone number.
    - c. Search all customers in a given city.  
(Use function overloading)
  8. Write a C++ program which will find the maximum of 3 integer numbers and maximum of 3 float numbers using function overloading.
  9. Create a C++ class my date with three members dd, mm, yy. Write a menu driven program with the following options.
    - Increment date by 1 day.
    - Subtract 2 days from date.

10. Write a C++ program using class to calculate square and cube of given number using inline function.
11. Write a C++ program using class to check maximum of two integer numbers using Inline function and conditional operator.
12. Create two classes' dist1 (meters, centimeters) and dist2 (feet, inches). Accept two distances from the user, one in meters and centimeters and the other in feet and inches. Find the sum and difference of the two distances. Display the result in both (meters and centimeters) as well as feet and inches (use friend function).
13. Write a class matrix which stores a matrix of integers of given size. Write necessary member functions accept and display Matrix.
14. Write a friend function for square matrices which interchanges the elements of m<sup>th</sup> and n<sup>th</sup> column.

**Topic: Default Argument, Static member functions and call by reference**

1. Write a C++ program using class to calculate simple interest amount use default value for rate.
2. Write a C++ program using class called 'Clock' that has hours, minute and second as integer data members. The class has the following member functions:

void setclock(int, int, int )	to set the initial time of clock object.
void showclock()	to display time in hh:min:sec format.

Write a function tick( ) which by default increment the value of second by 1 or according to user specified second. The clock uses 24 hours format.

3. Write a C++ program to define function power to raise a number m to a power n the function takes a double value for m. and integer value for n and return the result correctly use a default value of 2 for n to make the function calculate squares when this argument is omitted.
4. Write a C++ program using class with function replace (char\* str, char c1, char c2) every occurrence of c1 in str should be replaced with c2 and return number of replacement it makes use default value for char c2.
5. Write a C++ program to create a class Employee with data members as Emp\_Name, No\_of\_Days\_worked, Pay\_Rate\_Per\_Day. Write necessary member functions to calculate and display the salary of Employee. (Use default value for Pay\_Rate\_Per\_Day)

6. Write a C++ program to create a class Novel which contains data members as N\_Id, N\_Name, N\_Author. Write member functions to accept and display Novel information also display Count of novels. (Use Static data member to maintain Count of novels)
7. Write a C++ program to create a class Item with data members Item\_Code, Item\_Name, Item\_Price. Write member functions to accept and display Item information also display number of objects created for a class. (Use Static data member and Static member function)
8. Design a class which contain static data member and member function Print() which displays number of times print function is performed irrespective of the object responsible for print using static data member.
9. Write a C++ program to create a class which contains two data members. Write member functions to accept, display and swap two entered numbers using call by reference.

### **Topic:-Constructor and Destructor**

1. Write a C++ program to create a class which contains single dimensional integer array of given size. Write a member function to display sum of given array elements. (Use Dynamic Constructor to allocate and Destructor to free memory of an object)
2. Write a C++ program using class which contains two data members of type integer. Create and initialize the object using default constructor, parameterized constructor and parameterized constructor with default value. Write a member function to display maximum from given two numbers for all objects
3. Write a C++ program to create a class Part which contains data members as Part\_Id, Part\_Name, Part\_Price. Create and Initialize all values of Part object by using parameterized constructor and copy constructor. Display the values of Part object. (Part\_price should be right justified with a precision of two digits)
4. Write a C++ program to create a class Date which contains three data members as dd, mm, yyyy. Create and initialize the object by using parameterized constructor and display date in dd-mon-yyyy format. (Input: 29-12-2016 Output: 29-Dec-2016) Perform validation for month.
5. Create a C++ class for a student object with the following attributes—roll no, name, number of subjects, marks of subjects. The number of subjects varies for each student. Write a parameterized constructor which initializes roll number, name and number of subject and creates the array for marks dynamically, write member function for accepting marks and display all information of student also display the student having the highest percentage.
6. Create a class FDAccount containing members as:

Fdno, Name, Amt, Interest\_rate, Maturity\_amt, Number\_of\_months

Use parameterized constructor to set appropriate details, where interest rate should be default argument. Calculate maturity amt using interest rate and display all the details.

7. Write a c++ program to implement class complex. Use default constructor to initialize 0 to both real & imaginary part. Use copy constructor .Write necessary member functions to accept, display, add & multiply two complex numbers.
8. Define a class string. Use different constructors and do the following
  - a. Create un-initialized string objects.
  - b. Create objects with string constants.
  - c. Concatenate two strings.
  - d. Display a desired string.

### **Topic: Operator Overloading**

1. Create a class time that contains hours, minute and seconds as data members. Write the member function to overload operator '+' to add two object of type time,(Use Parameterized constructor to accept values for time).
2. Create a class CString to represent a string.
  - A)Overload the + operator to concatenate two strings
  - B)Overload <=, == to compare 2 strings.
  - C)Overload - (unary) to change the case.
3. Create a class Rational to represent a Rational number. Write member functions to accept and display Rational numbers. Perform the Basic Arithmetic operation: Addition, Subtraction, Multiplication and Division using operator overloading for two Rational number.
4. Write a class matrix which stores a matrix of integers of given size. Write necessary member functions and overload the following operators.
  - (binary) subtract two matrices and stores the result in third.
  - + multiplies two matrices and stores the result in third.
  - !(equals to) returns 1 if two matrices are same.
5. Define a c++ class fraction

```
class fraction
{

long numerator;
    long
    denominator;
public:

fraction(long n=0,long d=0);
}
```

Overload the following operators as member or friend. Unary++(pre and post both)

Overload as friend functions operators << and >>.

6. Define a class for 3 dimensional points. Write necessary member functions to accept and display the point object.

Overload the following operators:

<b>Operator</b>	<b>Example</b>	<b>Purpose</b>
* (Binary)	p3=p1*p2	multiplies coordinates of point p1 with p2.
- (Binary)	p3=p2-p1	subtracts coordinates of p1 from p2.

7. Create a class Array which contains

- int \*ptr

- int n

Write a menu driven program :

- to Accept an array from user

- to display it.

- to add two arrays using operator overloading.

- to subtract two arrays using operator overloading.

8. Create a class currency containing rupees and paiseas data members. Write necessary member functions using operator overloading for the following:

1. currency (long int rup=0,int paise=0)

C currency

2. &operator += (currency &) ( to add one currency to another)

3. currency & ( to subtract one currency from

1. operator - = (currency &) another)

Accept Rupee & paise from user and display it.

9. Consider a class Matrix Class Matrix

```
{  
int a[3][3];  
Public:
```

```
//methods;
```

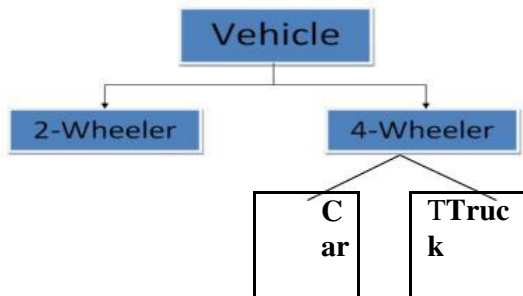
```
};
```

Overload the - (Unary) should negate the numbers stored in the object.

### **Topic: Inheritance**

1. Create base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called cylinder and rectangle from the base shape. Add to the base class, a member function get\_data(), print\_data() to initialize base class data members and display\_area(), display\_perimeter() to compute and display area and perimeter of shape.

2. Design the classes using following hierarchical inheritance



Each class has constructor for initialization and display function. Write a program to accept details of n cars and display the details.

3. Implement multiple and hierarchical inheritance. The class Allrounder derives information from both Bowler and Batsman classes which in turn derive information from the class Cricketer. Define all four classes and write a program to create, update and display information contained in Allrounder objects display the object having the highest number of runs.

### Topic: Virtual Function

- Q.1)** Create a base class Shape. Derive three different classes Circle, Rectangle and Triangle from Shape class. Write a C++ program to calculate area of Circle, Rectangle and Triangle. (Use pure virtual function).
- Q.2)** Create a base class Conversion. Derive three different classes Weight (Gram, Kilogram), Volume (Milliliter, Liter), Currency (Rupees, Paise) from Conversion class.

Write a C++ program to perform read, convert and display operations. (Use Pure virtual function)

- Q.3)** Create a base class Media. Derive two different classes Book (Book\_id, Book\_name, Publication, Author, Book\_price) and CD (CD\_title, CD\_price, CD\_capacity) from Media. Write a C++ program to accept and display information of both Book and CD. (Use pure virtual function)
- Q.4)** Create a base class Roundshape(radius). Derive three different shapes as Circle, Sphere and cylinder(height) from Roundshape.

```

Class Roundshape
{
    Protected : float
    radius; Public:
    static float Pi;

    Roundshape(float); //default
    argument virtual float area( )=0;
};

```

Write a C++ program to calculate area of Circle, Sphere and cylinder.

### Topic: File Handling

- Q.1)** Write a C++ program to read the contents of a text file. Count and display number of characters, words and lines from a file. Find the number of occurrences of a given word present in a file.
- Q.2)** Write a C++ program to read the contents of a “Sample.txt” file. Store all the uppercase characters in”Upper.txt”, lowercase characters in ”Lower.txt” and digits in “Digit.txt” files. Change the case of each character from “Sample.txt” and store it in “Convert.txt” file.
- Q.3)** Write a C++ program to create a class Department which contains data members as Dept\_Id, Dept\_Name, H.O.D., Number\_Of\_staff. Write necessary member functions to
- i. Accept details from user for ‘n’ departments and write it in a file “Dept.txt”.
  - ii. Display details of department from a file.
  - iii. Count the number of objects stored in a file.

- Q.4)** Create a C++ class MyFile containing:

```

- fstream f;
- char *filename;

```

Write necessary member Functions using operator overloading: << To display the contents of a file  
>> To write the contents into a file

- Q.5)** Write a C++ program which will accept ‘n’ integers from user through command line argument. Store Prime numbers in file “Prime.txt” and remaining numbers in “Others.txt”.

**Q.6)** Write a program in C++ that copies one file to another by changing the case of every alphabet in the file. Also count the total number of characters, words and lines in the file.

**Q.7)** Create a class Distance containing Feet and Inches as data members. Write a C++ program to read distance from user, store it in the file and to read it from the file and display it to the user.

Use operator overloading for the following:

1. <<to write distance object in inches format to a file
2. >>to read inches from File.

**Q.8)** Create a C++ class MyFile containing data member:

Fstream fp;

Char \*fn;

Write necessary member Functions using operator overloading:

- 1.+ F3=F1+F2 Put contents of F1 and F2 in F3.
- 2.- -F3 Changes the case of all upper and lower case characters in F3.

**Q.9)** Create a C++ class Student having data members:

-RollNo

-Name

-Marks

Write necessary member functions:

1. to accept the details and store it into the file "school.dat"
2. to read the details from file and display it.
3. to update a given record into the file.

**Q.10)** Create a C++ class Medicalshopee containing

-medicine\_no

-medicine\_name

-qty

-price

Medicine details are stored into the file "medical.txt". When any medicine has to be sold, it is first searched into the file, if found, the qty is decremented by that much qty to be sold.

**Q. 11)** Write a C++ program to implement the following:

Define a class 'city' with data members name and STD code. Accept 'n' cities with STD codes from user. Store this data in the file 'cities.txt'. Write a program that reads the data from file cities.txt into the array. Output the list of city with STD codes. Write a search function to search a city by name and display its STD code.



**Q. 12)** Assuming that a text file named FIRST.TXT contains some text written into it, write a function named copy\_upper(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT contains all words from the file FIRST.TXT in uppercase.

**Q. 13)** Assuming that a text file named FIRST.TXT contains some text written into it, write a function named vowelwords(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT, to contain only those words from the file FIRST.TXT which start with a vowel.

**Q.14)** Create a C++ class MyFile containing data  
member: fstream fp;  
Char \*fn;

Write member function for the following:  
void display() // reads the files

void append(); // append a character to a file  
void fclose(); // close the file

**Q.15)** Write a C++ program to create a text file which stores employee (emp\_id,emp\_name,emp\_sal) information. Write a menu driven program with the options

- Append
- Modify
- Display
- Exit

**Q.16)** A file contains a list of person name and their telephone numbers in the following form: Ajay 12345  
Vijay 98765

The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read the file and output the list in two columns. Implement the following tasks.

- Determine the telephone number of the specified person.
- Determine the name if a telephone number is known.
- Update the telephone number, whenever there is a change.

## **Topic: Template**

**Q.1)** Write a program in C++ to perform the following using the function template concepts:

- To read a set of integer numbers
- To read a set of float numbers
- Find out average of the integer numbers
- Find out average of the float numbers

**Q.2)** Define a class template Array to represent an array of any type and size. Define appropriate constructors for the class.

Define member functions to Accept and display an Array object.

Find the largest element in the array object.

Example: `Array<int, 10> a; //creates an array of the type int and size 10.`

**Q.3)** Using Standard Template Library(STL), write a program to obtain the list of files present in current directory and display these files in sorted order of name, type and size.

**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)


Course Code: BCA-406  
Total Contact Hours: 48 hrs.  
Total Marks: 100

Course Title: LAB II (Web Technology )  
Total Credits: 04

**Note that these are only sample assignments. Teachers may conduct practical's by preparing similar types of examples**

**Sample Assignments on HTML And CSS**

- 1) Create an html page with following specifications
  - a) Title should be about MYFAMILY
  - b) Put image of your sweet home in the background
  - c) Place address at the bottom in smaller size
  - d) Add names of family members in a different color, style and typeface
  - e) Add scrolling text with a message of your choice
  - f) Add family photo at the top.  
(Use appropriate header and footer for the page)
- 2) Create an html page with following specifications
  - a) Title should be about MYINDIA
  - b) Place country map in the background.
  - c) Add at least 5 state names each in a different color, style and typeface
  - d) Add images of each state at the bottom.
- 3) Write HTML code to design a web page displaying list of hyperlinks. Divide the browser screen into two frames. The frame on the left will be an image. The frame on the right will be a menu consisting of hyperlinks. Clicking on any one of these hyperlinks will display related information as a new page which must be open in frame on the right hand side.

<b>Welcome to Savitribai Phule Pune University</b>	
	<ul style="list-style-type: none"><li>• <a href="#"><u>About Us</u></a></li><li>• <a href="#"><u>Placement cell</u></a></li><li>• <a href="#"><u>Sports</u></a></li></ul>

4) Write HTML code to display following train details using table.

Train name	Starting place	Destination place	Time		Fare
			Arrival	Departure	

(Insert at least 5 records)

5) Write HTML and CSS code to design a web page. Divide the browser screen into two frames. The first frame will display the heading. Divide the second frame into two columns that will display the information as shown below.

Use External style sheet with appropriate attributes to display information of each frame.

Department of Science	
<p><b>Subjects for different Semester of B.C.A.(Sci.) course</b></p> <ul style="list-style-type: none"> <li>• <u>F. Y. B. C. A (Sci.)</u></li> <li>• <u>S. Y. B. C. A (Sci.)</u></li> <li>• <u>T. Y. B. C. A (Sci.)</u></li> </ul>	<p style="text-align: center;"><u>F. Y. B. C. A (Sci.)</u></p> <ul style="list-style-type: none"> <li>• <b>Sem I</b> Fundamentals of Computer Basic C Programming</li> <li>• <b>Sem II</b> Advance C DBMS</li> </ul>

6) Write HTML and CSS code to design a web page. Divide the browser screen into two frames. The first frame will display the heading. Divide the second frame into two columns. The first column will display hyperlinks. After clicking on each hyperlink related information should be displayed in the second column in table format.

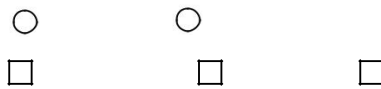
Use style sheet with appropriate attributes to display information of each frame.

Country	List of the Players	
	Name of the Game	List of the Players
	1 . <u>India</u>	
	2 . <u>England</u>	Cricket
3 . <u>Australia</u>		
	Tennis	Saniya Mirza Soniya Nehval Mahesh Bhupati

### Sports Details

7) Write HTML code to design the following form.

<u>Personal Information</u>			
First Name :	<input type="text"/>		
Last Name :	<input type="text"/>		
Address :	<input type="text"/>		
Mobile Number	<input type="text"/>		
Gender	Male	Female	
Your Interests	Computer	Sports	Music
<input type="button" value="Submit"/>		<input type="button" value="Reset"/>	



8) Write HTML to design the following output.

**User Information**

Name :

Gender :  Male     Female

Age :

▼

Hobbies :

- 9) Create an html page with following specifications
- Title should be about COLLEGE WEB SITE . Put the windows Logo image in the background Place College name at the top of the page in large text.
  - Add names of courses offered each in a different color, style and typeface.
  - Add scrolling text with a welcome message.
  - Add college image at the bottom. (Use external CSS)

10) Write HTML and CSS code to design a web page displaying list of hyperlinks. Divide the browser screen into two frames. The first frame will display the heading. Divide the second frame into two columns. The frame on the left will be a menu consisting of hyperlinks. Clicking on any one of these hyperlinks will display related information as a new page which must be open in frame on the right hand side.

Use Inline style sheet with appropriate attributes to display information of each frame.

### IT Industries in INDIA

City	Pune
1. <a href="#">Pune</a>	<ul style="list-style-type: none"> <li>• Infosys</li> <li>• TCS</li> <li>• Tech-Mahindra</li> <li>• Persistent</li> </ul>
2. <a href="#">Bangalore</a>	
3. <a href="#">Hyderabad</a>	
4. <a href="#">Delhi</a>	

11) Write HTML code to design a website for Online Shopping. Design two pages which contains hyperlink to each other. All pages must have different background.

Divide the first home page horizontally into two frames. The first frame contains logo and name of the shopping site. Divide the second frame vertically in two frames out of which First(right) frame displays two hyperlink images such as Mobiles, Car and second(left) frame displays information about shopping site in scrolling form.

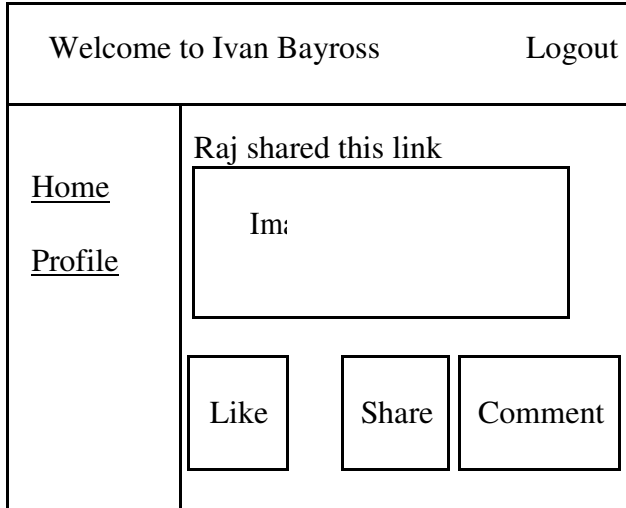
Divide the second page vertically in two frames. The first frame displays Category list (which are again hyperlinked) clicking on any one of these hyperlinks will display related information (Image, Name of the Model & Price) of the category in a second frame.

12) Write HTML code to design a website for Flight Reservation.

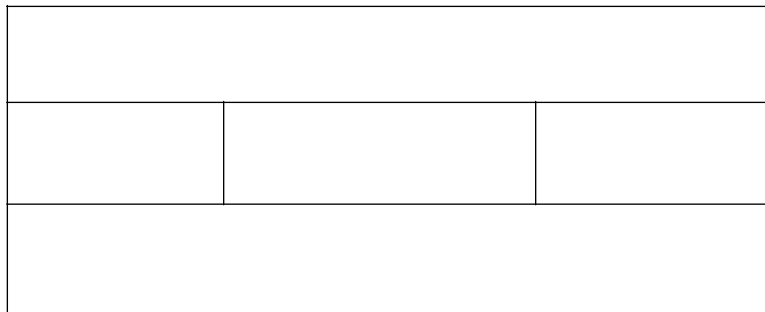
Logo	<b>AirIndia</b>	Indi a	▽
<p align="center">Manage Your Home Trip</p>			
<p align="center">Booking Information</p>			
<p>From : <input type="text"/></p>			
<p>To : <input type="text"/> <input type="text"/></p>			
<p>Departure Date : <input type="text"/></p>			
<p>Returning Date : <input type="text"/></p>			
<p align="center">E</p>			
<p>Adults Children economy</p>			
<p>1 to <input type="text"/> Executive ▾</p>			
<p>9 ▾ 0 to 9 ▾ First</p>			
<p align="center">Book Now</p>			



13) Write HTML code to design a website of Facebook. Design



14) Write HTML code to divide the web page into different frames and each frame will display the text as shown below.



15) Write HTML code to display calendar of the current month.

- **Introduction to JavaScript**

- 1) Write JavaScript Program to accept number from user and check whether it is Palindrome or not.
- 2) Write a JavaScript program that accepts a string from user. pass this string as parameter to a function name 'check\_vowel' on button click event and return the count of number of vowels within the string.
- 3) Write JavaScript Program to accept 1 to 7 numbers from user and display appropriate Weak of the day.
- 4) Write a JavaScript code for counting odd, even, prime numbers from 1 to100
- 5) Write a JavaScript program to read employee details and generate pay slip which will calculate net salary of an employee.

**Payslip**

<b>Name of Employee:</b>	<input type="text"/>
<b>Department</b>	<input type="text"/>
<b>Designation:</b>	<input type="text"/>
<b>Basic Salary</b>	<input type="text"/>
<b>HRA</b>	<input type="text"/>
<b>DA</b>	<input type="text"/>

- **Function in JavaScript**

- 1) Write JavaScript function to display Clock in the textbox.
- 2) Write a JavaScript function to accept First Name and Last Name from user. Display users entered information.

[Hint: use alert and prompt dialog box]

- 3) Write a JavaScript program to Display current Day, Date, Month, Year and Time on the web page and greet the user accordingly.

- **Validation in JavaScript**

- 1) Write a JavaScript to accept email address from user and count number of @ occurrences and number of Dot (.) occurrences.
- 2) Write a JavaScript code for accepting name and mobile number from user. (use validations).
- 3) Required Validations are: Name text field should not accept numbers, special characters, alphanumeric characters, length should be 20 characters.
- 4) Mobile number text field should not accept characters, special characters, alphanumeric characters, length should be 10 digits
- 5) Write a JavaScript code for accepting user name and password (use validations).
- 6) Required Validations are: User name text field should not accept numbers, special characters, alphanumeric characters, length should be 20 characters.
- 7) Password text field length should not be less than 3 characters.
- 8) Write a JavaScript code for calculating bills. If quantity is less than 5 then there is any discount, If quantity is greater than 5 and less than 25 then discount is 5%, If quantity is greater than 25 and less than 50 then discount is 15%, If quantity is greater than 50 discount is 20%.
- 9) Write a JavaScript code to accept number from user. Make all validation and print that number in reverse order.
- 10) Write a JavaScript code to calculate maximum, minimum, sum and average of numbers in an array. Make all validation.

- 11) Write a JavaScript program to design student registration form and perform following validation:
- Check all fields should not contain a null value
  - Check name field contains only alphabets
  - Mobile No. field should be of 10 digits.
  - Pin code field should be of 06 digits.

## Student Registration form

Name of Student:	<input type="text"/>
Mobile No.	<input type="text"/>
Address Line:	<input type="text"/>
City:	<input type="text"/>
State	<input type="text"/>
Pincode	<input type="text"/>
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

- **Introduction to PHP**

- 1) Design a HTML form to accept two numbers from the user and write a PHP script to display multiplication table of that range in tabular format.
- 2) Design a HTML form to accept a number from the user and write a PHP script to display digits of a number in word.(e.g 123 one two three)
- 3) Design a HTML page for the following (Accept input from user, use radio buttons)
  - a. Area of circle
  - b. Circumference of Circle
  - c. Volume of Sphere.
- 4) Write a PHP script to Perform selected action.(use switch case)
- 5) Design a HTML page for the following (Accept input from user, use radio buttons)
  - a. Area of square

- b. Area of Rectangle
  - c. Area of Triangle
- 6) Write a PHP script to Perform selected action.(use switch case)
  - 7) Design a HTML page an integer number and write a PHP script to count the number of digits and also calculate sum of digit in the number.
  - 8) Write a PHP script for the following: Design a form to accept two strings. Compare them using (=, ==, === operator). Append second string at the end of first string.
  - 9) Design a HTML page to check for the following (Accept input number from user, use radio buttons)
    - a. Prime
    - b. Armstrong
    - c. Perfect
  - 10) Write a PHP script to perform selected action (use switch case)
  - 11) Write a PHP script to change Background color of the browser using switch statement according to a day of the week.
  - 12) Write a PHP script to set selected image from Combo Box (DropDownList) to the background of Page.

- **Array in PHP**

- 1) Write a PHP script to define an associative array for the New user login with the details: Username, Password, Age, Qualification and Phone number. Create an associative array to store the user input and display it on other form. Check the validations for no field are left blank and age not less than 12. Design necessary screen layouts for the same.  
**Concept Used:** Associative array, form validations.
- 2) Write a menu driven program to perform the following operations on an associative array(Use Radio Buttons , use pre defined Array)
  - a. Display the elements of an array along with the keys.
  - b. Display the size of an array
  - c. Delete an element from an array from the given key/index.
- 3) Write a menu driven program to perform the following operations on an associative array(Use Radio Buttons , use pre defined Array)
  - a) Reverse the order of each element's key-value pair [use array\_flip()].

- b) Traverse the elements in an array in random order [use Iterator function].
  - c) Sort the array by values.
- 4) Accept a string from the user and display letter for given position number (Using array built-in functions).
- 5) Declare a Multidimensional Array. Display specific element from a Multidimensional array. Also delete given element from the Multidimensional array.(After each operation display array content [Hint : use print\_r() ]
- 6) Define an array. Find the elements from the array that matches the given value using appropriate search function.
- 7) Write a menu driven program to perform the following stack and queue related operations:[Hint: use Array\_push(), Array\_pop(), Array\_shift(), Array\_unshift() ]
- a) Insert an element in stack
  - b)Delete an element from stack
  - c)Display the contents of stack
- 8) Write a menu driven program to perform the following operations on associative arrays:
- a) Sort the array by values in ascending, descending order.
  - b) Sort the array by values without changing the keys.
  - c) Filter the odd elements from an array.
- 9) Write a menu driven program to perform the following operations on associative arrays:
- a) Merge the given arrays.
  - b) Find the intersection of two arrays.
  - c) Find the union of two arrays
  - d) Find set difference of two arrays.
- 10) Design a HTML page to accept elements of two matrices of same order (use two text box) and write a PHP script to display addition and multiplication.(use array\_chunk)
- 11) Write a menu driven program to perform the following stack and queue related operations:[Hint: use Array\_push(), Array\_pop(), Array\_shift(), Array\_unshift() ]
- a) Insert an element in queue

b) Delete an element from queue

12) Design a HTML page to accept string and write a PHP script to filter all vowels from the string. (use Array\_filter)

• **Function in PHP**

- 1) Design a HTML form to accept a number from the user and write a PHP script to find Factorial of a number using user defined function.
- 2) Design a HTML form to accept three numbers from the user and write a PHP script to find largest of those three numbers using Anonymous function.
- 3) Design a form to accept the input values from the user for Bank Loan and write a PHP script using functions to calculate the EMI amount, Number of EMIs and the final amount of payment (addition of all EMIs).

Bank Loan	
Loan A_C.No	<input type="text"/>
Loan-Amt	<input type="text"/>
Interest	<input type="text"/>
<input type="text"/>	<input type="text"/>

EMI Plann	
Loan amount	: _____
EMI Amount:	_____ -

- 4) Write a PHP script for the following: Design No.of a EMI: form to accept a string. Write a function to count the total number of vowels (a, e, i, o, u) from the Period Final Amount: string. Show the occurrences of each vowel from the string. Check whether the given string is a palindrome or not, without using built-in function. (Use radio buttons and the concept of function. Use 'include' construct or Submit Reset require stmt.)
- 5) Write a PHP script for the following: Design a form to accept two numbers from the user. Give options to choose the arithmetic operation (use radio buttons). Display the result on the next form. (Use the concept of function and default parameters. Use 'include' construct or require stmt)

- 6) Write a PHP script for the following: Design a form to accept the details of 5 different items, such as item code, item name, units sold, rate. Display the bill in the tabular format. Use only 4 text boxes. (Hint : Use of explode function.)
- 7) Write a PHP script for the following
  - a. design a form to accept two numbers form user
  - b. Give option to choose an arithmetic operation (use radio Button)
  - c. Display the result on next form
  - d. use the concept of default parameters
- 8) Write a PHP script to change the preferences of web page like font size, font style, font color. Display the selected setting on next page
- 9) Design a HTML form to accept two strings from the user. Write a PHP function to find whether the small string appears at the start of the large string. Provide a text box to accept the string that will replace all occurrences of small string present in the large string. Also split the large string into separate words. (Use regular expressions)
- 10) Design a HTML form to accept two strings from the user. Write a PHP function to find the first occurrence and the last occurrence of the small string in the large string. Also count the total number of occurrences of small string in the large string. Provide a text box to accept a string, which will replace the small string in the large string. (Use built-in functions)

Design a HTML form to accept two numbers from the user. Give options to choose the arithmetic operation (use radio buttons). Write a PHP function to display the result on the next form. (Use the concept of function and default parameters)



**Second Year B.C.A. (Science) Semester IV**  
(To be implemented from Academic year 2017-18)

**Course Code: BCA 407**  
**Total Contact Hours: 24 hrs. (30 Lectures)**  
**Total Marks: 50**

**Course Title: Grid and Cloud Computing**  
**Total Credits: 02**  
**Teaching Scheme: Theory- 03 Lect./Week**

Unit No.	Content	No.of Lectures
1	<p><b>Grid Computing – An overview [Book 1]</b></p> <ul style="list-style-type: none"> <li>• High-Performance Computing</li> <li>• Cluster Computing</li> <li>• Peer-to-Peer Computing</li> <li>• Internet Computing</li> <li>• Grid Computing - What Grid Computing Is, Peer-to-Peer Networks and Grid Computing, Cluster Computing and Grid Computing , Internet Computing and Grid Computing</li> <li>• Grid Computing Models</li> <li>• Open Grid Services Architecture</li> <li>• Types of Grids - Departmental Grids, Enterprise Grids, Extraprise Grids, Global Grids, Compute Grids, Data Grids, Utility Grids</li> <li>• Grid Networks -Grid Network Peering Points</li> <li>• Grid Applications Characteristics</li> </ul>	12
2	<p><b>Benefits of Grid Computing [Book 2]</b></p> <ul style="list-style-type: none"> <li>• Exploiting underutilized resources</li> <li>• Parallel CPU capacity</li> <li>• Virtual resources and virtual organizations for collaboration</li> <li>• Access to additional resources</li> <li>• Resource balancing</li> <li>• Reliability</li> <li>• Management</li> </ul>	3
3	<p><b>Cloud Computing – A overview [Book 3]</b></p> <ul style="list-style-type: none"> <li>• Defining Cloud Computing</li> <li>• The NIST model</li> <li>• Deployment models – Public, Private, Hybrid</li> <li>• Service models – Infrastructure as a Service (IaaS), Platform as a service (PaaS), Software as a Service (SaaS), Cloud reference model.</li> <li>• Examining the characteristics of Cloud Computing</li> <li>• Benefits of Cloud Computing</li> </ul>	11

	<ul style="list-style-type: none"> <li>• Disadvantages of Cloud Computing</li> </ul>	
4	<p><b>Abstraction and Virtualization</b> [Book 3]</p> <ul style="list-style-type: none"> <li>• Using Virtualization Technology</li> <li>• Load Balancing and Virtualization – The Google Cloud</li> <li>• Understating Hypervisors – Virtual Machine types</li> <li>• Exploring SaaS – salesforce.com, PaaS- force.com, IaaS – Amazon EC2</li> </ul>	4

**Reference Books:**

- 1) Grid Computing : A practical guide to technology and applications – Ahmar Abbas, Charles River Media Inc.
- 2) Introduction to Grid Computing – Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi. IBM International Technical Support Organization. [ibm.com/redbooks](http://ibm.com/redbooks).
- 3) Cloud Computing Bible – Barrie Sosinsky. Willey India Edition.
- 4) Cloud Computing Principles and Paradigms- Rajkumar Buya, James Broberg, Andrzej Goscinski. Willey publication.
- 5) Grid Computing – Joshy Joseph, Craig Fellenstein, Pearson Education.