Post Graduate Department of Computer Sciences,
The University of Kashmir,
Srinagar - 190006

Choice Based Credit System Curriculum for

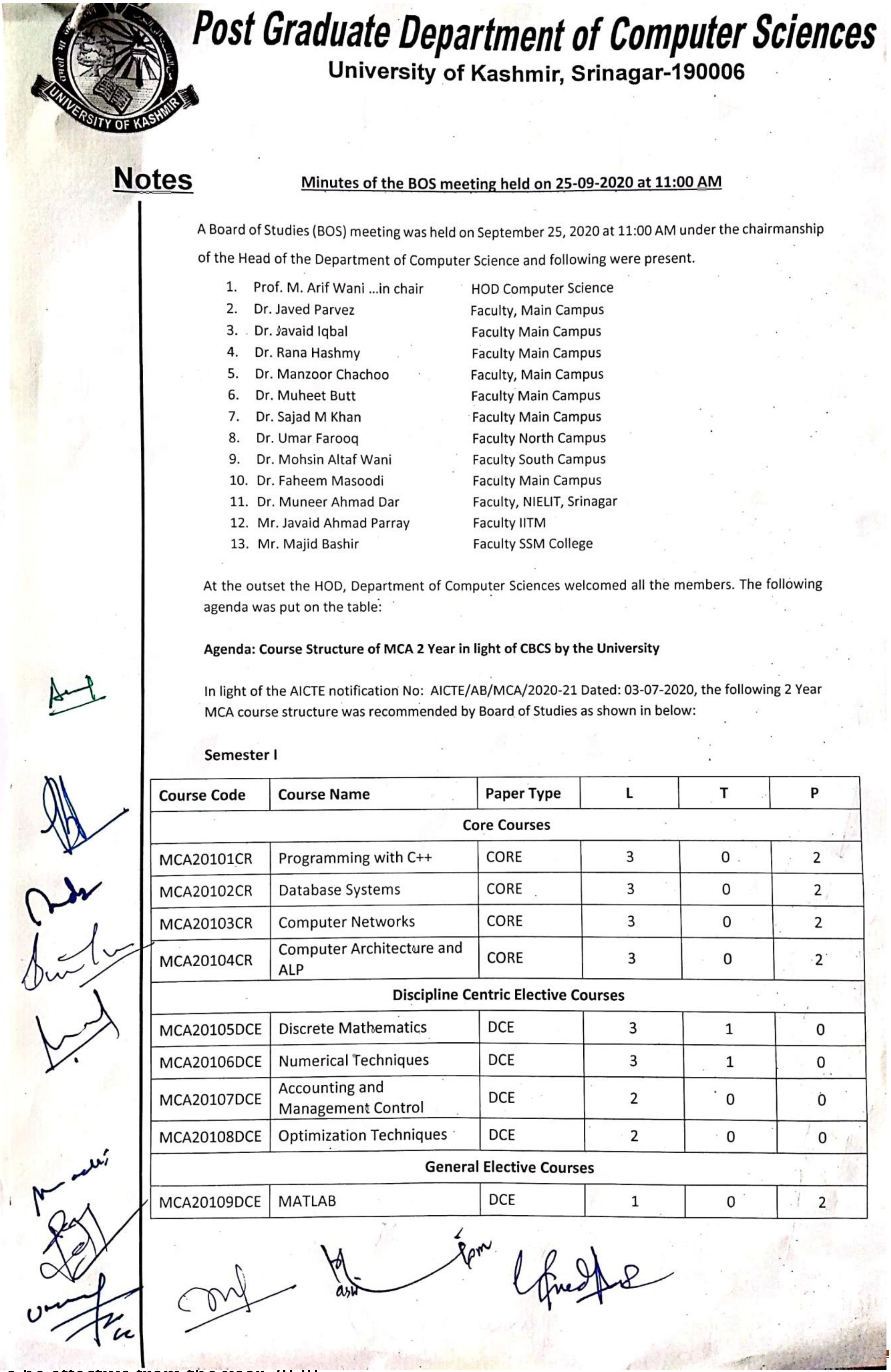
Master of Computer Applications (MCA) Programme 2020 – 2022

Eligibility for 2-year MCA degree Programme:

“Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree.

OR

Passed B.Sc./ B.Com./ B.A with Mathematics at 10+2 Level or at Graduation Level Obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying Examination”



Post Graduate Department of Computer Sciences

University of Kashmir, Srinagar-190006

Notes

Minutes of the BOS meeting held on 25-09-2020 at 11:00 AM

A Board of Studies (BOS) meeting was held on September 25, 2020 at 11:00 AM under the chairmanship of the Head of the Department of Computer Science and following were present.

1.	Prof. M. Arif Wani ...in chair	HOD Computer Science
2.	Dr. Javed Parvez	Faculty, Main Campus
3.	Dr. Javaid Iqbal	Faculty Main Campus
4.	Dr. Rana Hashmy	Faculty Main Campus
5.	Dr. Manzoor Chachoo	Faculty, Main Campus
6.	Dr. Muheet Butt	Faculty Main Campus
7.	Dr. Sajad M Khan	Faculty Main Campus
8.	Dr. Umar Farooq	Faculty North Campus
9.	Dr. Mohsin Altaf Wani	Faculty South Campus
10.	Dr. Faheem Masoodi	Faculty Main Campus
11.	Dr. Muneer Ahmad Dar	Faculty, NIELIT, Srinagar
12.	Mr. Javaid Ahmad Parray	Faculty IITM
13.	Mr. Majid Bashir	Faculty SSM College

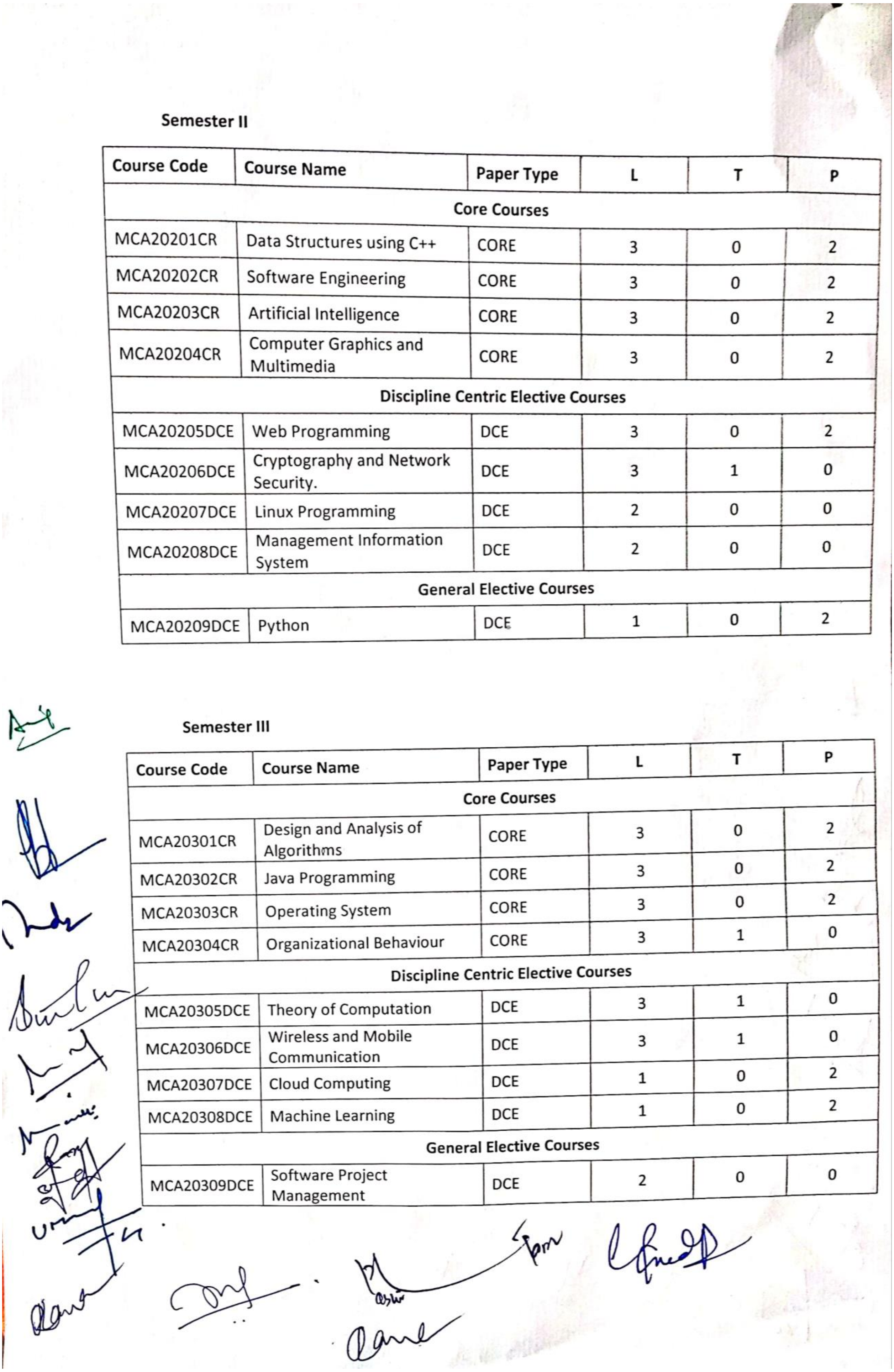
At the outset the HOD, Department of Computer Sciences welcomed all the members. The following agenda was put on the table:

Agenda: Course Structure of MCA 2 Year in light of CBCS by the University

In light of the AICTE notification No: AICTE/AB/MCA/2020-21 Dated: 03-07-2020, the following 2 Year MCA course structure was recommended by Board of Studies as shown in below:

Semester I

Course Code	Course Name	Paper Type	L	T	P
Core Courses					
MCA20101CR	Programming with C++	CORE	3	0	2
MCA20102CR	Database Systems	CORE	3	0	2
MCA20103CR	Computer Networks	CORE	3	0	2
MCA20104CR	Computer Architecture and ALP	CORE	3	0	2
Discipline Centric Elective Courses					
MCA20105DCE	Discrete Mathematics	DCE	3	1	0
MCA20106DCE	Numerical Techniques	DCE	3	1	0
MCA20107DCE	Accounting and Management Control	DCE	2	0	0
MCA20108DCE	Optimization Techniques	DCE	2	0	0
General Elective Courses					
MCA20109DCE	MATLAB	DCE	1	0	2

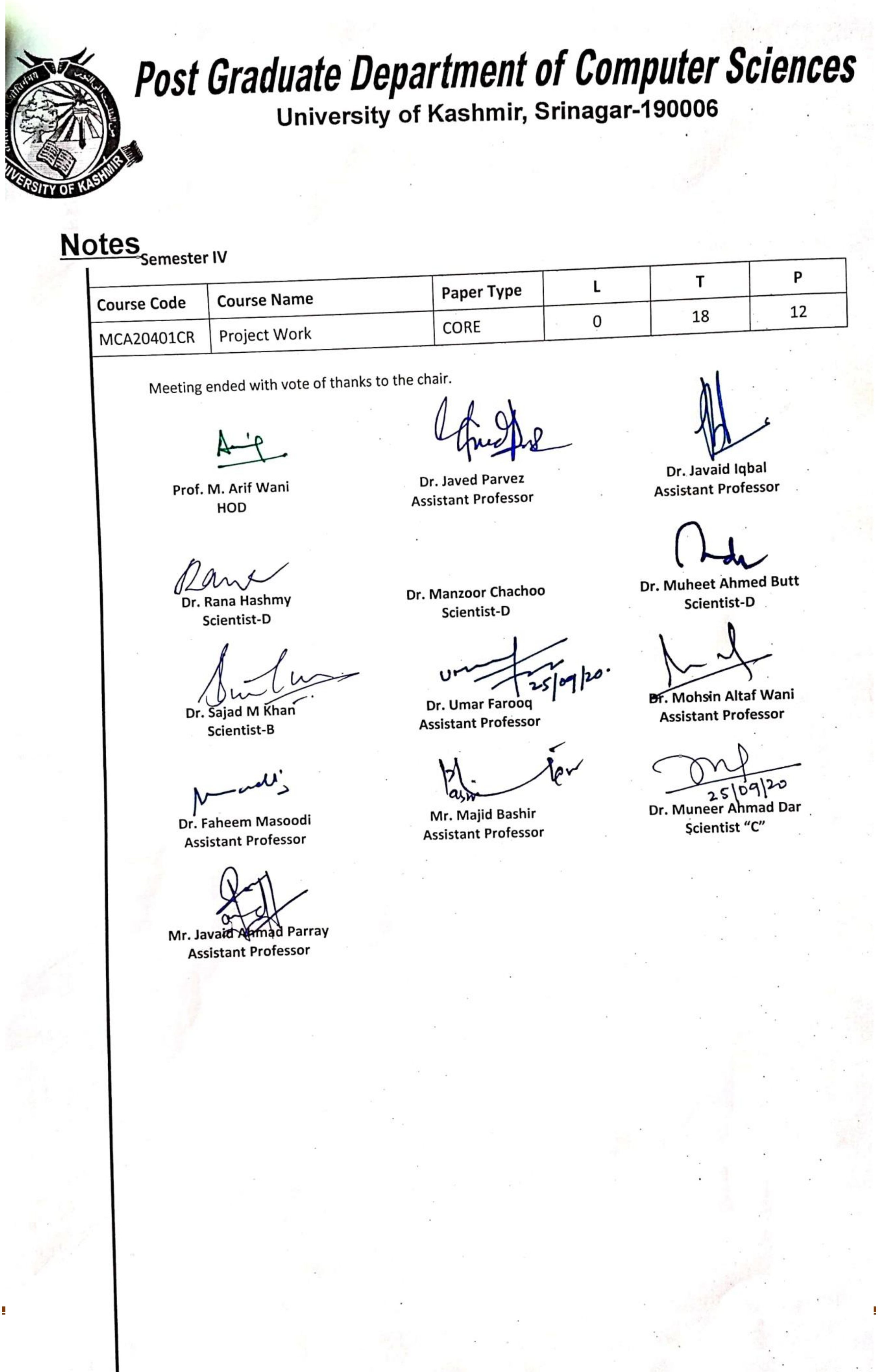


Semester II

Course Code	Course Name	Paper Type	L	T	P
Core Courses					
MCA20201CR	Data Structures using C++	CORE	3	0	2
MCA20202CR	Software Engineering	CORE	3	0	2
MCA20203CR	Artificial Intelligence	CORE	3	0	2
MCA20204CR	Computer Graphics and Multimedia	CORE	3	0	2
Discipline Centric Elective Courses					
MCA20205DCE	Web Programming	DCE	3	0	2
MCA20206DCE	Cryptography and Network Security.	DCE	3	1	0
MCA20207DCE	Linux Programming	DCE	2	0	0
MCA20208DCE	Management Information System	DCE	2	0	0
General Elective Courses					
MCA20209DCE	Python	DCE	1	0	2

Semester III

Course Code	Course Name	Paper Type	L	T	P
Core Courses					
MCA20301CR	Design and Analysis of Algorithms	CORE	3	0	2
MCA20302CR	Java Programming	CORE	3	0	2
MCA20303CR	Operating System	CORE	3	0	2
MCA20304CR	Organizational Behaviour	CORE	3	1	0
Discipline Centric Elective Courses					
MCA20305DCE	Theory of Computation	DCE	3	1	0
MCA20306DCE	Wireless and Mobile Communication	DCE	3	1	0
MCA20307DCE	Cloud Computing	DCE	1	0	2
MCA20308DCE	Machine Learning	DCE	1	0	2
General Elective Courses					
MCA20309DCE	Software Project Management	DCE	2	0	0



Post Graduate Department of Computer Sciences

University of Kashmir, Srinagar-190006

Notes

Semester IV

Course Code	Course Name	Paper Type	L	T	P
MCA20401CR	Project Work	CORE	0	18	12

Meeting ended with vote of thanks to the chair.

Prof. M. Arif Wani
HOD

Dr. Javed Parvez
Assistant Professor

Dr. Javaid Iqbal
Assistant Professor

Dr. Rana Hashmy
Scientist-D

Dr. Manzoor Chachoo
Scientist-D

Dr. Muheet Ahmed Butt
Scientist-D

Dr. Sajad M Khan
Scientist-B

25/09/20.
Dr. Umar Farooq
Assistant Professor

Dr. Mohsin Altaf Wani
Assistant Professor

Dr. Faheem Masoodi
Assistant Professor

Mr. Majid Bashir
Assistant Professor

25/09/20
Dr. Muneer Ahmad Dar
Scientist "C"

Mr. Javaid Ahmad Parray
Assistant Professor

Semester I

Subject Code: MCA20101CR
Subject Name: Programming with C++

Unit I [12 L]
Data Types, Identifiers, Variables Constants and Literals. Arithmetic Relational Logical and Bitwise. Basic input/output statements **[2L]**
Control structures: if-else statement, Nested if statement, Switch statement Loops: while loop, do while, for loop, Nested loops**. [3L]**
Arrays**:** Declaration; initialization; 2-dimensional and 3-dimensional array, passing array to function, Strings and String functions, and character arrays. **[3L]**
Functions; prototype, passing parameters, storage classes, identifier visibility, Recursive functions[4L]

Unit II [12L]
Command-line arguments. Variable scope, lifetime. Multi-file programming, Introduction to macros. File processing in C. [4L]
Structures and unions: syntax and use, members, structures as function arguments passing structures and their arrays as arguments [2L]
Pointers: variables, pointers and arrays, pointers to pointers, strings, pointer arithmetic, portability issues, pointers to functions, void pointers, pointer to structure. [4L]
Introduction to object oriented programming, Abstraction, Encapsulation [2L]

Unit III [8L]
Introduction to classes and objects; Access specifiers, Constructor; destructor; Function overloading; Operator overloading; friend functions; Use of call-by-reference for efficiency. Copy constructor. [4L]
Inheritance: Single, Multiple, and Multilevel Inheritance[2L]
Virtual functions and Polymorphism/Dynamic binding vs Static binding; Virtual Destructors[2L]

Unit IV [8 L]
Pure virtual function; concrete implementation of virtual functions[2L]
Templates: Function Templates, Class Templates, Member Function Template and Template Arguments, namespaces, Exception Handling Concepts[4L]
Input and Output: Streams classes, Stream Errors, Disk File I/O with streams. [2L]

Reference Books:

1. FOSTER AND FOSTER “C by discovery” RRI penram.
2. ROBERT LAFORE “Object orientation with C++ Programming” Waite Group.
3. YASHWANT KANETKAR “Let us C” PHI.
4. E. BALAGURUSWAMI “Programming in ANSI C” Tata McGraw Hill.
5. BJARNE STROUSTRUP “The C++ programming language” Pearson Education.
6. HERBERT SCHILD “C++ The complete Reference” Tata McGraw Hill.

Subject Name: C / C++ Lab

Week 1

Q1. Write a program to demonstrate the use of Output statements that draws any object of your choice e.g. Christmas Tree using '*'
Q2. Write a program that reads in a month number and outputs the month name.
Q3. Write a program that demonstrate the use of various input statements like getchar(), getch(), scanf().
Q4. Write a program to demonstrate the overflow and underflow of various datatype and their resolution?
Q5. Write a program to demonstrate the precedence of various operators.

Week 2.

Q1. Write a program to generate a sequence of numbers in both ascending and descending order.
Q2. Write a program to generate pascals triangle.
Q3. Write a program to reverse the digits of a given number. For example, the number 9876 should be returned as 6789
Q4. Write a program to convert an amount (upto billion) in figures to equivalent amount in words.
Q5. Write a program to find sum of all prime numbers between 100 and 500.

Week 3

Q1. Create a one dimensional array of characters and store a string inside it by reading from standard input. Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number. Print the number and its frequency in a tabular form.
Q2. Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number. Print the number and its frequency in a tabular form.
Q3. Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.
Q4. Write a program which will arrange the positive and negative numbers in a one-dimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers. Example:

Original array contains: 10-15,1,3,-2,0,-2,-3,2,-9
Modified array: 10,1,3,0,2-15,-2,-2,-3,-9

Q5. Write a program to compute addition multiplication and transpose of a 2-D array

Week 4

Q1. Write a function to find the GCD and LCM of two numbers.
Q2. Implement a swap () function which exchanges the values of two integers. Call the function from the main to test the function with different values.
Q3. Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.
Q4. Write a function to generate the Fibonacci series using recursions?
Q5. Write a recursive function that adds first 'n' natural numbers?
Q6. Write a recursive function that finds factorial of a number?
Q7. Write a program to demonstrate the use of recursion in Tower of Hanoi problem.

Week 5.

Q1. Implement a program which uses multiple files for holding multiple functions which are compiled separately, linked together and called by main(). Use static and extern variables in these files.
Q2. Implement a function which receiver a pointer to a Student struct and sets the values of its fields.
Q3. Write a program which takes five arguments on command line, opens a file and writes one argument per line in that file and closes the file.
Q4. Write a program which creates Student (struct) objects using malloc and stores their pointers in an

array. It must free the objects after printing their contents.

Q5. Write a function char* stuff(char* s1,char* s2,int sp, intrp) to stuff string s2 in string s1 at position sp, replacing rp number of characters (rp may be zero).

Q6. Write a program to input name, address and telephone number of 'n' persons (n<=20). Sort according to the name as a primary key and address as the secondary key. Print the sorted telephone directory.

Week 6.

Q1. Write a program to find the number of occurrences of a word in a sentence ?

Q2. Write a program to concatenate two strings without using the inbuilt function?

Q3. Write a program to check if two strings are same or not?

Q4. Write a program to check whether a string is a palindrome or not?

Q5. Write a program to find the number of vowels and consonants in a sentence?

Q6. Write a program that reverse the contents of a string?

Week 7.

Q1. Write a program to demonstrate the array indexing using pointers.

Q2. Write a program to pass a pointer to a structure as a parameter to a function and return back a pointer to structure to the calling function after modifying the members of the structure?

Q3. Write a program to demonstrate the use of pointer to a pointer.

Q4. Write a program to demonstrate the use of pointer to a function.

Q5. Write a program to demonstrate the swapping the fields of two structures using pointers?

Week 8.

Q1. Write a program in C++ to define class ***complex*** which having two data members viz***real*** and ***imaginary***part ?

Q2. Write a program in C++ to define class ***Person*** which having multiple data members for storing the different details of the person e.g. name,age, address, height etc.

Q3. Write a program to instantiate the objects of the class ***person*** and class ***complex*** ?

Q4. Write a C++ program to add member function that displays the contents of class person and class complex?

Q5. Write a C++ program to demonstrate the use of scope resolution operator?

Week 9

Q1. Write a program in C++ which creates objects of Student class using default, overloaded and copy constructors.

Q2. Write a program to demonstrate the use of different access specifiers.

Q3. Write a C++ program to demonstrate the use of inline, friend functions and ***this*** keyword.

Q4. Write a C++ program to show the use of destructors.

Q5. Write a program in C++ demonstrates the use of function overloading.

Week 10

Q1. Write a C++ program to overload the '+' operator so that it can add two matrices .

Q2. Write a C++ program to overload the assignment operator.

Q3. Write a C++ program to overload comparison operator operator== and operator!= .

Q4. Write a C++ program to overload the unary operator.

Week 11

Q1. Write a program in C++ which creates a single-inheritance hierarchy of Person, Employee and Teacher classes and creates instances of each class using new and stores them in an array of Person * .

Q2. Write a program in C++ which creates a multiple-inheritance hierarchy of Teacher classes derived from both Person, Employee classes. Each class must implement a Show() member function and utilize scope-resolution operator

Q3. Write a program in C++ demonstrates the concept of function overriding?

Q4. Write a C++ program to show inheritance using different levels?

Week 12

Q1. Write a C++ program to demonstrate the concepts of abstract class and inner class?

Q2. Write a C++ program to demonstrate the use of virtual functions and polymorphism?

Q3. Write a C++ program to demonstrate the use of pure virtual functions and virtual destructors?

Q4. Write a C++ program to swap data using function templates.

Q5. Write a C++ program to create a simple calculator which can add, subtract, multiply and divide two numbers using class template.

Week 13

Q1. Write a C++ program to demonstrate the concept of exception handling.

Q2. Write a C++ program to create a custom exception.

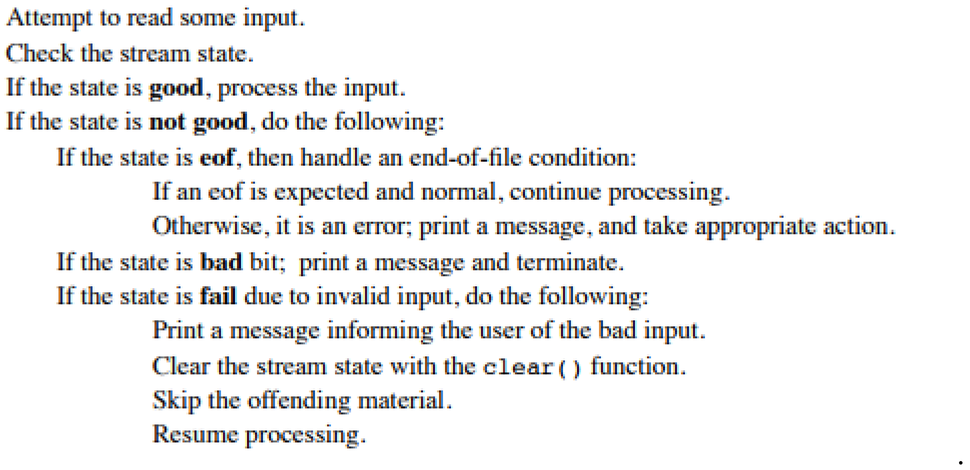
Q3. Define a class with appropriate data members and member functions which opens an input and output file, checks each one for being open, and then reads name, age, salary of a person from the input file and stores the information in an object, increases the salary by a bonus of 10% and then writes the person object to the output file. It continues until the input stream is no longer good.

Q4.. Define a class with appropriate data members and member functions for testing all the error bits with member functions.

This example has a get_int function that tries to read an integer from file, and checks everything. The function has three reference-type parameters: an input stream to read from, a bool good_flag parameter, which will be set to true if a valid integer was read in, and an integer variable, in which the read-in value will be stored. It returns a bool value that is true if we should continue reading from the stream, and false if we should stop. Function main() calls get_int in the condition of a while loop which continues as long as get_int returns true. When get_int returns, if value_is_good is true, then datum contains an integer that was successfully read in. We print it, and then continue to the loop. When get_int returns false, we stop looping and terminate.

Inside get_int, we first attempt to read an integer. If the stream is good, the variable contains a valid value, and we can continue reading the stream. If eof was encountered, a valid variable value was not obtained, and it time to stop reading the input. Likewise, if there was a "hard I/O" error. Finally, if the input was invalid, a valid value was not obtained. The example follows a policy to skip the rest of the line is skipped, and we should continue trying to read the stream (unless we hit end of file while skipping the rest of the line).

The pattern represented by this example is :



```
Attempt to read some input.
Check the stream state.
If the state is good, process the input.
If the state is not good, do the following:
    If the state is eof, then handle an end-of-file condition:
            If an eof is expected and normal, continue processing.
            Otherwise, it is an error; print a message, and take appropriate action.
    If the state is bad bit;  print a message and terminate.
    If the state is fail due to invalid input, do the following:
            Print a message informing the user of the bad input.
            Clear the stream state with the clear() function.
            Skip the offending material.
            Resume processing.
```

Subject Name: C / C++ Tutorials

Week 1

Q1. C++ program to print Right-angled pyramid of numbers as shown below:

```
   1
  2 3
 4 5 6
7 8 9 10
```

Q2. C++ program to keep calculating the sum of digits of a number until the number is a single number.
Q3. C++ program to skip some of the array elements.
Q4. C++ program to print your name randomly on the screen with coloured text.

Week 2

Q1. C++ program to find the last index of a character in a string.
Q2. C++ program to get weekday from a given date.
Q3. C++ program to get previous date of a given date.
Q4. C++ program to check that the given string is numeric or not.

Week 3

Q1. C++ program to check whether the given date is in valid format or not.
Q2. C++ program to Fibonacci number using different methods (Recursion method, Dynamic programming method, without using array method).
Q3. C++ program to convert a number to word (upto four digits only).
Q4. C++ program to find factorial of large numbers using arrays.

Week 4

Q1. C++ program to use function as a LVALUE using reference variable.
Q2. C++ program to demonstrate methods of passing in function.
Q3. C++ program to demonstrate function overloading.
Q4. C++ program to generate random numbers.

Week 5

Q1. C++ program to check a year for leap year.
Q2. C++ program to add two integer numbers using pointers.
Q3. C++ program to create a class to read and add two distances.
Q4. C++ program to create a class for student to get and print details of a student.

Week 6

Q1. C++ program to demonstrate example of array of objects.
Q2. C++ program to demonstrate example of friend function with class.
Q3. C++ program to create a class to read time in HH:MM:SS format and display into seconds.
Q4. C++ program to create a class to read time in seconds and convert into time in HH:MM:SS format.

Week 7

Q1. C++ program to demonstrate example of default constructor or no argument constructor.
Q2. C++ program to demonstrate example of parameterized constructor.
Q3. C++ program to demonstrate example of constructor overloading.
Q4. C++ program to demonstrate example of constructor using *this* pointer.

Week 8

Q1. C++ program to demonstrate example of simple inheritance.
Q2. C++ program to read and print student’s information using two classes and simple inheritance.
Q3. C++ program to demonstrate example of multilevel inheritance.
Q4. C++ program to demonstrate example of multiple inheritance.

Week 9

Q1. C++ program to demonstrate example of Hierarchal inheritance to get square and cube of a number.
Q2. C++ program to read and print employee information with *deptt* and *pf* information using Hierarchal inheritance.
Q3. C++ program to demonstrate example of private simple inheritance.
Q4. C++ program to read and print employee information using Multiple inheritance.

Week 10

Q1. C++ program for unary minus (-) operator overloading.
Q2. C++ program for unary increment (++) and decrement (--) ioperator overloading.
Q3. C++ program to add two objects using binary plus (+) operator overloading.
Q4. C++ program to add two distances using binary plus (+) operator overloading.

Week 11

Q1. C++ program to demonstrate the concept of exception handling.
Q2. C++ program to create a custom exception.
Q3. C++ program to swap data using function templates.
Q4. C++ program to create a simple calculator which can add, subtract, multiply and divide two numbers using class template.

Week 12

Q1. C++ program to create a file.
Q2. C++ program to create a file to write and read text in/from file.
Q3. C++ program to write and read time in/from a binary file using *fstream*.
Q4. C++ program to demonstrate the use of virtual functions and polymorphism.

Subject Code: – MCA20102CR
Subject Name: Database Systems

Unit I

Database basics – DBMS, RDBMS, ORDBMS. Characteristics and Services of Database Management System, Advantages and Disadvantages of DBMS. Schemas, and Instances, Comparison Three Schema Architecture and Data Independence. Database Languages. DBMS architectures, Criterion for Good Database Design [3**L**]

Data Models: Object Based Model, Record Based Model. Codds rules [3**L**]. Relational Algebra – Basic and Derived operators with examples [2**L**]

Database Design through Functional Dependencies & Normalization. Functional Dependencies, Lossless Join, Normal Forms: 1NF, 2NF, 3NF, 4NF (BCNF) [**4L**]

Unit II

Introduction to Oracle, Tools of Oracle, Introduction to SQL, Data Types, Data Definition Language, Data Manipulation Language [**4L**]

Transaction Control Language, Integrity Constraints, SQL Functions, Set Operators and Joins, View, Synonym and Index [**4L**]

Sub Queries and Database Objects, User Management, Privileges, Locks and SQL Formatting Commands. [**4L**]

Unit III

Introduction to PLSQL, Basic Architectures, Data Types, Conditional and Looping Logic [**4L**].

Concept of Implicit and Explicit Exception Handling, Cursors and Database Triggers, Subprograms and Packages. [**4L**]

UNIT IV

Transaction Processing –Transaction Processing Basics, Transaction and Systems Concepts, Transaction Properties. [**2L**]

Characterizing Schedules and Recoverability, Schedules and Serializability. [**2L**]

Concurrency Control - Two Phase Locking, Timestamp Ordering. [**2L**]

Database Recovery – Basic Concepts, Transaction Rollback, Recovery based on Deferred and Immediate Update, Shadow Paging [**2L**]

Reference Books:

1. Fundamentals of Database Systems, Elmasri&Navathe, Pearson Education Ltd. 2016.
2. Database System Concepts, Silberschatz, Korth, &Sudarshan, McGraw-Hill, 2011.
3. An introduction to Database Systems, Date C. J, Pearson, 2006.
4. SQL, Pl/SQL: The Programming Language of Oracle, Bayross I. BPB Publications, 2009
5. Teach Yourself SQL and PL/SQL Using Oracle 8i and 9i with Sqlj, Bayross I. BPB Publications, 2003

Subject Name: Database Systems (Tutorials)

Week 1
Q1. Explain the terms Data, Database and DBMS?
Q2. What are the various characteristics of Database Management Systems?
Q3. Explain basic concepts of ER model in DBMS.
Q4. Explain Codd's rule for relational DBMS.

Week 2
Q1. Explain basic concepts of RDBMS and its ER model.
Q2. . Explain various normal forms in DBMS.
Q3. Explain Relational Algebra and Relational Calculus.
Q4. Explain different types of Database Keys.

Week 3
Q1. Explain basic concepts of SQL (DDL, DML, TCL, DCL and DQL).
Q2. Practice "create" command of DDL on tables.
Q3. Practice "Alter" command of DDL on tables.
Q4. Practice "Drop, Rename and truncate" commands of DDL on tables.

Week 3
Q1. Practice "select" command of DML on tables.
Q2. Practice "insert" command of DML on tables.
Q3. Practice "update" command of DML on tables.
Q4. Practice "delete" command of DML on tables.

Week 4
Q1. Practice queries Using Operators: Arithmetic, Logical and Comparison.

Q2. Practice "Commit" command of TCL.
Q3. Practice "Commit work" command of TCL.
Q4. Practice "save point and Rollback" command of TCL.

Week 5
Q1. Practice "grant and revoke" commands of DCL.
Q2. Practice "Where and like" clause on tables.
Q3. Practice "order by and group by" clause on tables.
Q4. Practice "Having" clause on tables.

Week 6
Q1. Practice applying Data Integrity constraints (Domain Constraint, Validity Integrity and Entity Integrity) on tables.
Q2. Practice applying Referential Integrity constraints on tables.
Q3. Practice various Aggregate functions of SQL.
Q4. Practice Date Functions, Numeric Functions, Character Functions, Conversion Functions of SQL.

Week 7
Q1. Practice SQL Set operations on tables (Union, Union All, Intersect, Minus).
Q2. Practice "Cross Join" on tables.
Q3. Practice "Inner Join or Equi Join" on tables.
Q4. Practice "Natural Join" on tables.

Week 8
Q1. Explain PLSQL and its basic architecture.
Q2. Explain various Data types in PLSQL.
Q3. Explain various concepts of Exception handling in PLSQL.
Q4. What are triggers in PLSQL
. **Week 9**
Q1. Explain packages in PLSQL.
Q2. Explain Cursors in PLSQL
Q3. Explain the concept of functions in PLSQL.
Q4. Explain various loops used in PLSQL.
Week 10
Q1. What is a Transaction and explain its properties.
Q2. Explain various states of Transactions.
Q3. Explain Serializability.
Q4. What are Equivalence Schedules?
. **Week 10**
Q1. Explain Recoverability.
Q2. What is concurrency control?
Q3. Explain Lock based protocols in concurrency control.
Q4. Explain Two Phased Locking Protocol.
. **Week 11**
Q1. Explain Time Stamp Ordering protocol.
Q2. What is Thomas Write Rule?
Q3. What is Database Recovery?
Q4. What are various structures used for Database recovery?
. **Week 12**
Q1. Explain Forward and Backward Rolling.
Q2. Explain recovery based on deferred update.
Q3. Explain recovery based on immediate update
Q4. Explain recovery based on Shadow paging.

Subject Name: Database Systems Labs

Week 1

Data Definition Language: Create, Alter, Drop Commands on Tables, Users, Objects

Q1. Create table Student with following attributes and perform the following operations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS	ST_TELNO
Date Type	Number	Varchar	Char	Varchar2
Size	6	30	35	15

Table: 1.1 : Student Table

i. Add new attributes City, Street, Country with Datatype Varchar and length 30?

ii. Modify field ST_ROLLNO and change the size to 5?

iii. Remove column ST_ADDRESS?

iv. Describe the Table Student?

v. Drop Table Student?

vi. Copy Structure of one table to another

Q2. Create Users user1, user2, user3 and perform the following operations

i. Grant Session Privilege to the newly created users?

ii. Grant privileges for creating and manipulation tables?

iii. Grant data manipulation privileges to various users on tables?

iv. Grant privileges with grant option.

v. Revoke privileges.

Q3. Create Object ADDRESS and use the object in a Table DDL?

Week 2

Data Manipulation Language: Select, Insert, Update, Delete Command on Tables, Objects

Q1. Create table Student with following attributes and perform the following operations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_STREET	ST_CITY	ST_State	ST_Country	DTE_REG
Date Type	Number	Varchar	Char	Char	Varchar2	Varchar2	Date
Size	6	30	35	30	30	30	

Table 2.1: Student Table

i. Insert 10 records in the table 2.1

ii. Perform various Project Operations using Select Query.

iii. Perform various restrict operations using Select Query.

iv. Update records in the Table 2.1

v. Delete records in the Table 2.1

vi. Copy records from one Table STUDENT to other Table TEST? Assume both have same table structure.

Q2. Create table Student with ST_ADDRESS as Object Type with following attributes and perform the following operations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS				DTE_REG
			ST_STREET	ST_CITY	ST_State	ST_Country	
Date Type	Number	Varchar	Char	Char	Varchar2	Varchar2	Date
Size	6	30	35	30	30	30	

Table 2.2: Student Table

i) Insert 10 records in the table 2.2

ii) Perform various Project Operations using Select Query.

iii) Perform various restrict operations using Select Query.

iv) Update records in the Table 2.2

v) Delete records in the Table 2.2

vi) Copy records from one Table STUDENT to other Table TEST? Assume both have same table structure.

Week 3

Using Operators in SQL Plus: Arithmetic, Logical and Comparison

Q1. Create table STUDENT with following attributes and perform the following operations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_STREET	ST_CITY	ST_State	ST_Country	DTE_REG
Date Type	Number	Varchar	Char	Char	Varchar2	Varchar2	Date
Size	6	30	35	30	30	30	

Table 3.1: Student Table

i. Insert 10 records in the table 3.1

ii. Perform various Project Operations using Select Query.

iii. Perform various restrict operations using Select Query using various arithmetic and Logical Operators like

a. Less Than

b. Greater Than

c. Less Than or Equal to

d. Greater Than or Equal To

e. Equal to

f. Not Equal To

iv. Perform restrict operations shown to (iii) using various data types like numeric, Characters, Date.

v. Perform Update operations using various Arithmetic and Logical Operators on Table STUDENT

vi. Perform Delete operations using various Arithmetic and Logical Operators on Table STUDENT

vii. Use Insert and Select Commands together with Arithmetic and Logical Operators.

Transaction Control Language: Commit, Commit Work, Save Point and Rollback Commands

Q1. Perform Transaction Control Operations on the table 3.1 as under

1. Perform various data manipulation operations the table 3.1.
2. Create Five Savepoints from S1 to S5.
3. Rollback to Various savepoints and observe the changes in the table 3.1.
4. Perform various DDL operations the table and observe its effect on Savepoint and Rollback on the table 3.1.
5. Try to abnormally terminate the application to observe whether data is saved or not.
6. Use Commit and Commit Work commands to save the data permanently.

Week 4

Data Integrity: Domain Constraint (NOT NULL), Validity Integrity (CHECK), Entity Integrity (Primary and Unique Keys) , Referential Integrity (foreign Key), Column Level and Table Level constraints, Dropping Constraints.

Q1. Create table STUDENT with following attributes and perform various DML operations to verify Domain Constraint?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	NOT Null	Not NULL	NOT NULL

Table 4.1: Student Table for Verifying Domain Constraint

Q2. Create table STUDENT with following attributes and perform various DML operations to verify Validity Integrity?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	CHECK (ROLLNO > 20001 and ROLLNO < 30001	Not NULL	NOT NULL

Table 4.2: Student Table for Verifying Validity Integrity

Q3. Create table STUDENT with following attributes and perform various DML operations to verify Entity Integrity using Primary and Unique Keys?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	Primary/ Unique Keys	Not NULL	NOT NULL

Table 4.3: Student Table for Verifying Entity Integrity

Q4. Create table STUDENT with following attributes and perform various DML operations to verify Referential Integrity using Master and Detail Tables?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	Primary Key	Not NULL	NOT NULL

Table 4.4: Student Table for Verifying Entity Integrity (Parent Table)

Attribute Name	ST_ROLLNO	ST_DATE	ST_PRESENT_ABSENT
Date Type	Number	DATE	Varchar
Size	6		1 (A-ABSENT P-PRESENT)
Constraint	Foreign Key	Not NULL	NOT NULL

Table 4.5: Attendance Table for Verifying Referential Integrity (Child Table)

Q5. Also Perform below Mentioned Operations on tables using above Data Integrity Constraints.

i) Create System named Constraints and User Defined Constraints
ii) Create Column and Table Level Constraints?
iii) View User Constraints Table for checking Constraint Parameters?
iv) Create and Drop Constraints using alter command
v) Perform on_delete_Cascade and on_update_cascade

Week 5

SQL Plus Functions:

Single Row Functions: Date Functions, Numeric Functions, Character Functions, Conversion Functions, Miscellaneous Functions, Group Functions: Group by clause, Having Clause.

Q1. Check the Output of following SQL functions from Table 5.1 to 5.6.

Numeric Functions

Function Name	*Return Value*
ABS (x)	Absolute value of the number '*x*'
CEIL (x)	Integer value that is Greater than or equal to the number '*x*'
FLOOR (x)	Integer value that is Less than or equal to the number '*x*'
TRUNC (x, y)	Truncates value of number '*x*' up to '*y*' decimal places
ROUND (x, y)	Rounded off value of the number '*x*' up to the number '*y*' decimal places

Table 5.1: Numeric Functions explained

Character Functions

Function Name	*Return Value*
LOWER (string_value)	All the letters in *'string_value'* is converted to lowercase.
UPPER (string_value)	All the letters in *'string_value'* is converted to uppercase.
INITCAP (string_value)	All the letters in *'string_value'* is converted to mixed case.
LTRIM (string_value, trim_text)	All occurrences of *'trim_text'*is removed from the left of *'string_value'*.
RTRIM (string_value, trim_text)	All occurrences of *'trim_text'* is removed from the right of *'string_value'* .
TRIM (trim_text FROM string_value)	All occurrences of *'trim_text'* from the left and right of *'string_value'* ,*'trim_text'* can also be only one character long .
SUBSTR (string_value, m, n)	Returns *'n'* number of characters from *'string_value'* starting from the *'m'* position.
LENGTH (string_value)	Number of characters in *'string_value'* in returned.
LPAD (string_value, n, pad_value)	Returns *'string_value'* left-padded with *'pad_value'* . The length of the whole string will be of *'n'* characters.
RPAD (string_value, n, pad_value)	Returns *'string_value'* right-padded with *'pad_value'* . The length of the whole string will be of *'n'* characters.

Table 5.2: Character Functions explained

Date Functions

Function Name	*Return Value*
ADD_MONTHS (date, n)	Returns a date value after adding *'n'* months to the date *'x'*.
MONTHS_BETWEEN (x1, x2)	Returns the number of months between dates x1 and x2.
ROUND (x, date_format)	Returns the date *'x'* rounded off to the nearest century, year, month, date, hour, minute, or second as specified by the *'date_format'*.
TRUNC (x, date_format)	Returns the date *'x'* lesser than or equal to the nearest century, year, month, date, hour, minute, or second as specified by the 'date_format'.
NEXT_DAY (x, week_day)	Returns the next date of the *'week_day'* on or after the date *'x'* occurs.
LAST_DAY (x)	It is used to determine the number of days remaining in a month from the date *'x'* specified.
SYSDATE	Returns the systems current date and time.
NEW_TIME (x, zone1, zone2)	Returns the date and time in zone2 if date 'x' represents the time in zone1.

Table 5.3: Character functions explained

Conversion Functions

Function Name	*Return Value*
TO_CHAR (x [,y])	Converts Numeric and Date values to a character string value. It cannot be used for calculations since it is a string value.
TO_DATE (x [, date_format])	Converts a valid Numeric and Character values to a Date value. Date is formatted to the format specified by *'date_format'*.
NVL (x, y)	If *'x'* is NULL, replace it with *'y'*. *'x'* and *'y'* must be of the same datatype.
DECODE (a, b, c, d, e, default_value)	Checks the value of *'a'*, if *a = b*, then returns *'c'*. If *a = d*, then returns *'e'*. Else, returns *default_value*.

Table 5.4: Conversion functions explained

Miscellaneous Functions

Function Name	*Return Value*
Uid	returns the User ID

User name	Returns User Name
Vsize	Returns Number of Bytes in an Expression

Table 5.5: Miscellaneous functions explained

Group Functions

Function Name	*Return Value*
Avg	Returns Average of the Total
Max	Highest Value of the List
Min	Lowest Value of the List
Sum	Sum of the List
Count(*)	Returns Count of the rows

Table 5.6: Group functions explained

Q2. Using relevant table examples perform select operations using Group by and Having Clause.

Week 6

Set Operator and Joins: Union, Union All, Intersect, Minus. Joins: equi Join, Non Equi Join, Self Join, Outer Join.

Q1. Perform the following Set Operations on the Tables A and B shown below:

ST_ROLLNO	ST_NAME
101	Abid
102	Nasir
103	Tahir
104	Majid
105	Basid
106	Jehangir

Table 6.1: Table A

ST_ROLLNO	ST_NAME
105	Basid
106	Jehangir
107	Tameem
108	Sahir
109	Tabish
110	Faheem

Table 6.2: Table B

i) Union
ii) Union All
iii) Intersect
iv) Minus.

Q2. Using the Two Tables STUDENT & ATTENDANCE, insert data in the two tables and perform the following Join Operations shown below:

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	Primary Key	Not NULL	NOT NULL

Table 6.3: Student Table for Verifying Entity Integrity (Parent Table)

Attribute Name	ST_ROLLNO	ST_DATE	ST_PRESENT_ABSENT
Date Type	Number	DATE	Varchar
Size	6		1 (A-ABSENT P-PRESENT)
Constraint	Foreign Key	Not NULL	NOT NULL

Table 6.4: Attendance Table for Verifying Referential Integrity (Child Table)

i) Equi Join
ii) Non Equi Join
iii) Self Join
iv) Outer Join

Subject Code: MCA20103CR
Subject Name: Computer Networks

Unit I
Goals and applications of networks. LAN, MAN & WAN architectures. Concept of WAN subnet.**(4L)** Overview of existing networks. OSI Reference Model Architecture, TCP/IP Model and their comparison. **(3L)** Protocol layers and service models. OSI and Internet protocols.**(5L)**

Unit II
Internetworking concept and architectural model. Connection-oriented and connection-less approaches. Discuss ATM and Ethernet. Concept of Virtual Circuits, Concept of Autonomous systems and Internetwork Routing. Classful IP addresses. Subnetting, Subnet addressing, IP Multicasting.**(6L)** Internet Protocol (IP): connectionless delivery of datagrams (MTU, fragmentation, reassembly). IP header structure. IP Addressing. Efficiency and consistency trade-offs. **(6L)**

Unit III
Internet control protocols: ICMP, ARP and RARP. Concepts of delay, security, and Quality of Service (QoS). Reliable data transfer. Stop-and-Go evaluation. TCP and UCP semantics and syntax. TCP RTT estimation.**(4L)** Principles of congestion control. Principles of routing. Link-state and distance vector routing. Routing algorithms: Inter- and intra-domain routing. RIP, OSPF, BGP.CIDR. Transport Layer: UDP and TCP concepts. Socket API for Network Programming. **(8L)**

Unit IV
Client-Server application development using TCP & UDP sockets. Basic Server Architectures. Network Security: Overview of threats, cryptography, authentication, and firewalls their components. **(5L)** Encryption techniques and examples of encryption standards. Network management including SNMP. Network troubleshooting.**(7L)**

Reference Books:

1. Andrew Tanenbaum, “Computer Networks”, 4th Edition by Pearson.
2. Douglas Comer, “Internetworking with TCP/IP, Volume 1”, Pearson.
3. W. Richard Stevens, “UNIX Network Programming”, Pearson.
4. Maufer, “IP Fundamentals”, Pearson.
5. Douglas Comer, “Client-Server Programming with TCP/IP, Volume 3”, Pearson.

Subject Name: Computer Networks (Lab)

1. Network components such as Modem, Gateways, Routers, Switches, Cables etc.
2. Various network softwares, services and applications.
3. Network trouble shooting Techniques: Trouble shooting basic TCP/IP problems.
4. Commands like ipconfig, getmac, tracert, pathping, arp, ping, netstat, finger etc.
5. Straight cabling, Cross cabling, Signal testing, T568A and B wiring standards (including hands on practice)
6. Program that prints the address of www.bitmesra.ac.in
7. Program that prints all the addresses of www.indianrail.gov.in
8. Program that scans lower ports and prints them.
9. Program to list host names from command line, attempt to open socket to each one and print the remote host, the remote port, the local address and the local port.
10. Program for splitting the URLs entered into command line into component parts.
11. Program to list all the interfaces available on a workstation.

TCP/IP and UDP/IP socket Programming

1. Program for “echo” client. The Client enters data to the server, and the server echoes the data back to the clients.

2. Program for “echo” Server. The Server listens at the port specified and reads from client and echoes back the result.

Serial Port programming

1. Program to write out “Hello World” to a serial port or to a USB to Serial Converter.

Simple RPC Programming. (Introductory level)

Subject Code: MCA20104CR
Subject Name: Computer Architecture and ALP

Unit I:
8086 Microprocessor: 8086 Microprocessor Architecture (BIU, EU, Instruction Queue), Software Model
(General Purpose Registers, Segment Registers, Flag & Other Registers). Segmentation. [4L]
8086 Pin Functions, Minimum and Maximum Mode, The 8086 Memory System [2 L]
8086 Basic Programming: 8086 Programming Model, 8086 Instruction Formats, Addressing Modes. [4L]
The 8086 Instruction Set. [2L]
Unit II:
Assembly Language Programming: Significance, Assemblers and Linkers, TASM Directives – Data Definitions, Named-constants, User-defined, Segments, Subroutines, Macros, Modular-code. [2L]
Programming with Data Transfer, Arithmetic and Logical Instructions: Data Transfer, Arithmetic, Logical/Bit Manipulation, [4L]
Branching and Looping : Unconditional and Conditional Jump instructions, Decision making and looping, Loop instructions, ASCII and BCD Arithmetic, Processor Control Instructions. [4L]
Shift Instructions, Rotate Instructions and String Instructions [3L]
Unit III:
Stacks: Defining a stack, Push and Pop Instructions [2L]
Procedures: Defining and Calling procedure. CALL and RET instructions, Parameter Passing Methods, Far procedure [4L]
Macros: Working with macros, additional assembler directives [2L]
INT 21H: INT 21H Keyboard Services, Display Services, and File Manipulation Services. [3L]
Input/Output Instructions [2L]

Text Book: M.T. Savalia. 8086 Programming and Advanced Processor Architecture. Wiley India.
Reference Books:
1. T.P. Skinner. An Introduction to 8086/8088 Assembly Language Programming. John Wiley .
2. W. A. Triebel, A. Singh. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software,
Hardware, And Applications. Pearson Education, 2007.
3. B. B. Brey, The Intel Microprocessors: Architecture, Programming and Interfacing, Merrill, 2nd Edition, 1991.

Subject Name: Computer Architecture and Assembly Language Programming
Lab Work

List of experiments/practical-assignments for the course:

Week 1. This week students will learn how to declare, initialize and access varied-sized variables by using Assembler Directives and 8086 instructions.
a. Write a program that declares and initializes two integer variables (one 8-bit wide and another 16-bit wide), and then assigns new values to them using 8086 instructions.
b. Write a program that declares and initializes a String array (byte array) of 10 elements, and then assigns new values to each element individually using 8086 instructions.
Week 2. This week students will learn how to use INT 21H service to read integers and strings from keyboard and display them on screen.
a. Write a program that reads an integer value from keyboard using INT 21H keyboard service, stores it in memory, and displays it using INT 21H display service after doing necessary ASCII conversion.
b. Write a program that declares and initializes a String array (byte array), and uses INT 21H display service to displays all elements individually.
c. Write a program that declares and initializes two integer variables (one 8-bit wide and another 16-bit wide), and displays it using INT 21H display service after doing necessary ASCII conversion.
Week 3. This week students will learn how to perform addition and subtraction of 8-bit integer values.
a. Write a program that reads two 8-bit integers from keyboard (using INT 21H) and displays their sum (using INT 21H after doing necessary ASCII conversion).
b. Write a program that reads two 8-bit integers from keyboard (using INT 21H) and displays their difference (using INT 21H after doing necessary ASCII conversion).
Week 4. This week students will learn how to perform multiplication and division of 8-bit integer values.
a. Write a program that reads two 8-bit integers from keyboard (using INT 21H) and displays their multiplication result (using INT 21H after doing necessary ASCII conversion).
b. Write a program that reads two integers (16-bit and 8-bit) from keyboard (using INT 21H) and displays their division result (using INT 21H after doing necessary ASCII conversion).
Week 5. This week students will learn how to perform various arithmetic operations of 16-bit integer values.
a. Write a program that reads two 16-bit integers from keyboard (using INT 21H) and displays their sum (using INT 21H after doing necessary ASCII conversion).
b. Write a program that reads two 16-bit integers from keyboard (using INT 21H) and displays their difference (using INT 21H after doing necessary ASCII conversion).
c. Write a program that reads two 16-bit integers from keyboard (using INT 21H) and displays their multiplication result (using INT 21H after doing necessary ASCII conversion).
d. Write a program that reads two integers (32-bit and 16-bit) from keyboard (using INT 21H) and displays their division result (using INT 21H after doing necessary ASCII conversion).
Week 6. This week students will learn how to perform various bitwise operations on integer values.
a. Write to program reads two integers from keyboard (using INT 21H) and displays the result of the AND, OR, XOR, CMP and TEST operation (using INT 21H after doing necessary ASCII conversion).
b. Write to program reads two integers from keyboard (using INT 21H) and displays the result of the SHL, SHR, SAR, ROL, ROR, RCL, and RCR operation (using INT 21H after doing necessary ASCII conversion).
Week 7. This week students will learn how to use 8086 instructions for looping and decision making.
a. Write a program that reads an integer from keyboard (using INT 21H service), and iteratively

calculates its factorial.

b. Write a program that declares and initializes an array of 10 elements each 8-bit wide, reads an 8-bit integer from keyboard, searches its existence through the array, and displays the result of the search operation.

c. Write a program that declares and initializes an array of 10 elements each 8-bit wide, and sorts its elements in ascending order.

Week 8. This week students will learn how to use Macros and Conditional-Assembly.

a. Write a program that uses a Macro to exchange the values of two 16-bit integer variables.

b. Write a program that assembles code for division of two 16-bit integer variables if and only both integer values are non-zero.

Week 9. This week students will learn how to define a subroutine, pass parameters to it, and return value from it.

a. Write a program that defines a subroutine, which takes two 8-bit integers as parameters via Registers, calculates their sum, and returns the result to the caller.

b. Write a program that defines a subroutine, which takes two 8-bit integers as parameters via Stack, calculates their sum, and returns the result to the caller.

Week 10. This week students will learn how to use a subroutine to recursively solve a problem.

a. Write a program that defines a subroutine that uses recursion to calculate factorial of an integer read from keyboard.

Week 11. This week students will learn how to read and write files residing on secondary storage using INT 21H service.

a. Write a program that opens an existing text file in the current working directory, and display its contents.

b. Write a program that creates a file in current working directory, writes textual data to it (read from keyboard), and closes it.

c. Write a program that opens an existing text file in the current working directory, and appends textual data to it (read from keyboard), and closes it.

Week 12. This week students will learn how to write, install and use a custom software-interrupt.

a. Write a program that creates a subroutine to display “hello world!”, installs the subroutine as ISR, and subsequently uses it via INT interface.

b. Write a program that first creates a subroutine, which takes two 8-bit integers as parameters, does their summation and returns the result. Second, the program installs the subroutine as ISR, and subsequently uses it (via INT interface) to perform addition of two 8-bit integers read from keyboard.

Week 13. This week students will learn how to write a simple device driver for VGA.

a. Write a program that defines a subroutine, which takes three parameters – row, column, address of the String, and uses memory-mapped I/O to display it on screen. The program calls this subroutine to display a String inputted via keyboard.

b. Write a program that defines a subroutine, which takes three parameters – row, column, address of the String, and uses memory-mapped I/O to display it on screen. The program installs the subroutine as ISR and accesses it via INT interface to display a String inputted via keyboard.

Discipline Centric Elective Courses

Subject Code: MCA20105DCE
Subject Name: Discrete Mathematics

UNIT I

Proposition, Logic, Truth tables, Propositional Equivalence, Logical Equivalence, Predicates and Quantifiers; Sets: operations on sets, Computer representation of sets, Cardinality of a Set **(5L)**
Functions: Domain, Range, One-to-One, Onto, Inverses and Composition, Sequences and summations, The growth of functions. **(5L)**
Methods of Proof: Direct Proof, Indirect Proof, Mathematical Induction for proving algorithms; Counting techniques – Permutations, Combinations, The Pigeonhole Principle. **(5L)**

UNIT II

Discrete Probability, Advanced Counting Techniques: Inclusion-Exclusion, Applications of Inclusion-exclusion principle, recurrence relations, solving recurrence relation. **(5L)**
Relations: Relations and their properties, Binary Relations, Equivalence relations, Diagraphs, Matrix representation of relations and digraphs. **(5L)**
Computer representation of relations and digraphs; Transitive Closures, Warshall's Algorithm, Problem solving on Warshall's Algorithm.**(5L)**

UNIT III

Partially Ordered Sets (Posets), External elements of partially ordered sets, Hasse diagram of partially ordered set, isomorphic ordered set, Lattices: Properties of Lattices, complemented Lattices. **(5L)**
Graph theory: Introduction to graphs, Graph Terminology Weighted graphs, Representing Graphs, Connectivity of Graphs: Paths and Circuits, Eularian and Hamiltonian Paths, Matrix representation of graphs. Graph Coloring and its applications. **(5L)**

UNIT IV

Trees: Rooted trees, Application of trees: Binary Search Trees, Decision Trees, Prefix Codes, Tree traversal, trees and sorting, spanning trees, minimal spanning trees. **(5L)**
Finite Boolean algebra, Functions on Boolean algebra, Boolean functions as Boolean polynomials. Groups and applications: Subgroups, Semigroups, Monoids Isomorphism, Homomorphism. **(5L)**

Reference Books:

1. KENNETH H. ROSEN "Discrete Mathematics and Its Applications" The Random House/Birkhauser Mathematics series
2. LIU "Elements of Discrete Mathematics " Tata McGraw Hill
3. SCHAUMS "Discrete Mathematics " Tata McGraw Hill
4. KOLMAN/REHMAN "Discrete Mathematical Structures " Pearson Education
5. NICODEMI "Discrete Mathematics " CBS

Subject Name: Discrete Mathematics (Tutorials)

Tutorial Sheet #1 of Unit I

1) Determine whether following conditional statements are true or false. Also justify your answer:
   a. If 1+1 =3, then 2+2=4
   b. If 1+1 =3, then 2+2 = 5

2) Write the following propositions using p and q and logical connectives, where p and q are given to be:

   p: It is below freezing
   q: It is snowing
   1) If it is below freezing it is also snowing
   2) It is either below freezing, or it is snowing, but it is not snowing if it is below freezing

3) Prove that $(p\lor q) \land (\neg p)$ is a contingency

4) Find whether $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ is a tautology or a contradiction?

5) Show that:

   $\neg(p\lor (\neg p\land q))$ and $(\neg p \land \neg q)$ are logically equivalent by using the propositional laws

Tutorial Sheet #2 of unit I

1. Prove the following De-Morgan's law using truth table:

   $\neg(p\land q)$ is logically equivalent to $(\neg p \lor \neg q)$

2. Let P(x,yz) : " x + y = z". Find the truth values of the following:
   a. P(1, 2, 3)
   b. P(0, 0, 1)

3. Use quantifiers to express the statement:

   "Every student in this class has taken at least one course in Computer Science"

4. Let P(x,y) be the statement: "x + y = x – y ". If the universe of discourse for both x and y is the set of integers, what are the truth values of the following?
   a. For every x For some y P(x,y)
   b. For some x For some y P(x,y)

Tutorial Sheet #3 of Unit I

1) Let f be the function from

   Set A = { a, b, c, d } to Set B = { 1, 2, 3, 4},
   with f(a) = 4, f(b) = 2, f(c) = 1, f(d) = 3.
   Is f a bijection function? Justify your answer

2) Given:

   f(F) = (F – 32) * 5/9
   Make flow diagram to create the inverse of this function

3) Prove by Mathematical Induction that:

   $2 + 6 + 10 + \dots + (4n - 2) = 2n^2$

4) Find at least three sequences beginning with the term 1, 2, 4 whose terms are generated by a simple formula or rule.

5) How many students must be in a class to guarantee that at least two students receive the same score in the final exam, if the exam is graded on a scale from 0 to 100 points?

Tutorial Sheet #4 of Unit II

1. Each user on a computer has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?
2. Explain the counting process of finding the number of internet addresses available for computers on Internet.
3. What is the value of k after the following code has been executed?

```
k:=0
for $i_1$=1 to $n_1$
for $i_2$=1 to $n_2$
    :
    :
    :
for $i_m$=1 to $n_m$
k:=k+1
```

4. How many bit strings of length four do not have consecutive 1 s?
5. A playoff between two teams consists of at most five games. The first team that wins three games wins the playoff. In how many different ways can the playoff occur. Use tree diagram.

Tutorial Sheet #5 of Unit II

1. A young pair of rabbits (one of each sex) is placed on an island. A pair of rabbits does not breed until they are two months old. After they are two months old, each pair of rabbits produces another pair each month. Find a recurrence relation for the number of pairs of rabbits on the island after n months, assuming that no rabbits ever die.
2. Suppose that a person deposits $10,000 in a savings account at a bank yielding 11% per year with interest compounded annually. How much will be in account after 30 years?
3. Let H_n denote the number of moves needed to solve the Tower of Hanoi problem with n disks. Set up a recurrence relation for the sequence H_n
4. How many solutions does

   $X_1+x_2+x_3=11$

   have, where x_1,x_2,x_3 are nonnegative integers with $x_1<=3$, $x_2<=4$, $x_3<=6$?
5. How many onto functions are there from a set with six elements to a set with three elements?

Tutorial Sheet #6 of Unit II

1. Is the divides relation on the set of positive integers reflexive, symmetric, antisymmetric, and transitive?
2. How many reflexive relations are there on a set with n elements?
3. Let m be a positive integer with m>1. Show that the relation

   $R = \{(a,b)|a\equiv b(\text{mod } m)\}$

   is an equivalence relation on a set of integers.
4. What are the sets in the partition of the integers arising from congruence modulo 4?

Tutorial Sheet #7 of Unit III

1) Draw the Hasse diagram of (D(75), divides), where the set D(75) represents the set of all divisors of 75.
2) In the above question check whether (D(75), divides) forms a lattice or not.
3) Let S= {6,16,24,36,144}. Draw the Hasse diagram of (P(S), divides).

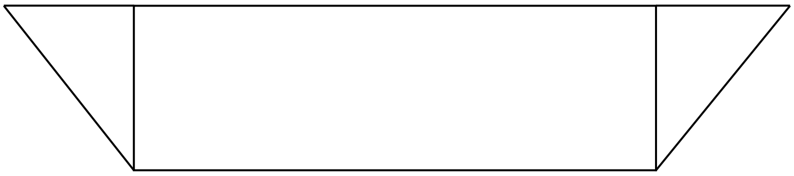
4) Find maximal and minimal elements of the following sets:
   a. Set A = { 3, 9, 12} with "Divides" relation
   b. Set B = { 1, 2, 4, 5, 10} with "Divides" relation

Tutorial Sheet #8 of Unit III

1) Draw a graph from following adjacency matrix with respect to ordering of vertices a, b, c, d:

$$\begin{array}{c|cccc} & \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ \hline \mathbf{a} & 0 & 1 & 1 & 0 \\ \mathbf{b} & 1 & 0 & 0 & 1 \\ \mathbf{c} & 1 & 0 & 0 & 1 \\ \mathbf{d} & 0 & 1 & 1 & 0 \end{array}$$

2) How many edges are there in a graph with 12 vertices, each of degree 4?
3) A connected graph has an Euler path but not an Euler circuit iff it has exactly two vertices of odd degree. Verify this theorem by drawing a graph of the said property.
4) Find the chromatic number of the following graph:



Tutorial Sheet #9 of Unit 4

1. Use Huffman coding to encode the following symbols with the frequencies listed:
   A: 0.08 B: 0.10 C: 0.12 D: 0.15 E: 0.20 F: 0.35
   What is the average number of bits used to encode the character?
2. What is the value of prefix expression **+ - * 2 3 5 / ↑ 2 3 4?**
3. What is the postfix form of the expression
   ((x + y) ↑ 2) + ((x-4)/3)

4. Find a binary search tree for the words
   mathematics physics geography zoology meteorology
   geology psychology chemistry
   using alphabetical order.

Tutorial Sheet #10 of Unit 4

1. Use Prim's algorithm to find a minimum spanning tree of any graph.
2. Suppose there are seven coins, all with the same weight, and a counterfeit coin that weighs less than the others. How many weighing's are necessary using a balance scale to determine which of the eight coins is the counterfeit one? Give an algorithm for finding this coin.
3. Show that the distributive law

   X (y + z) = xy + xz

is valid

4. Is the set Z, (a set of integers) monoid under usual operation of +, - ?

Subject Code: MCA20106DCE
Subject Name: Numerical Techniques

UNIT I

Computer Arithmetic: Introduction, Floating Point Representation and Arithmetic, Normalized Floating Point Representation of Numbers. **(3L)**
Approximations & Errors – Types of Programming Errors, Data Errors, Computer & Arithmetic Errors, Round off and Truncation Errors. Accuracy and Precision, Measures of Accuracy, Error Propagation.**(3L)**
Iterative Methods - Non-Linear Equations, Types of Methods to find solutions to nonlinear equations, Algorithms to Compute Roots of Equation – Methods of Tabulation or Brute Force Method, Method of Bisection, Secant Method, Newton-Raphson Method, Method for False Position.**(6L)**
Derivation of mathematical formulas and implementation of these methods.

UNIT II

Solution of Simultaneous Algebraic Equations: Linear Equations, Types of Methods to find solutions to linear equations. Algorithms to Solve Linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Gauss Seidel, L.U. Decomposition, Pivoting**(12L)**
Derivation of mathematical formulas and implementation of these methods.

Unit III

Interpolation: Lagrange Interpolated Polynomial, Newton's Methods of INTERPOLATION – Forward difference, Backward difference.**(4L)**
Least Square Approximation of Functions: Linear Regression and Polynomial Regression. **(4L)**
Taylor and Chebyshev Series **(4L)**
Derivation of mathematical formulas and implementation of these methods.

UNIT IV

Differentiation and Integration: Simpsons Rule, Trapezoidal Rule **(4L)**
Numerical solution of Differential Equations using Taylor Series Method, Euler's Method, Modified Euler's Method, Runge-Kutta Methods, Predictor Corrector Formula, Higher order Differential Equations, Comparison of Runge-Kutta and Predictor and Correction Methods.**(8L)**
Derivation of mathematical formulas and implementation of these methods.

Text Books

1. S.C.Chapra&R.P.Canale: "Numerical methods for Engineering". Tata McGraw Hill.
2. Krishenmurty and Sen : "Numerical Algorithms"
3. V. Rajaraman "Computer oriented numerical methods." Prentice Hall of India.
4. Grewal, B. S.: "Higher Engineering Mathematics", Hindustan Offset Problems Series.

Subject Name: Numerical Techniques (Tutorial)

WEEK 1:

Q 1: What is the difference between accuracy and precision? Define the two ways for measuring accuracy.
Q2: Define different types of errors.
Q3: Let X= 0.005998. Find relative error if x is truncated to 3 decimal digits
Q4: Let X= 0.005998. Find relative error if x is truncated to 3 decimal digits.
Q5: What is the difference between chopping and symmetric round-off. Illustrate with the help of examples.

WEEK 2:

Q1: What do you mean by approximation and error?
Q 2: Find the root of the equation 2x-x-3 = 0 graphically.
Q3: Find the root of the equation correct to three decimal digits using Bisection Method.
X3 – 2X - 5 =0
Q4: Write the programming implementation of Bisection method for Q3
Q5: Find the root of the equation correct to three decimal digits using False Position Method.
cos x -3x+1 = 0

WEEK 3:

Q1: Write the programming implementation of False Position method for Q4.
Q2: Find the root of the equation correct to three decimal digits using Secant Method.
cos x $-xe^x=0$
Q3: Write the programmatic implementation of the Q1
Q4: Find the root of the equation correct to three decimal digits using Newton Raphson Method.
$X^3 - 2X - 5 = 0$.
Q5: Write the programmatic implementation of the Q3.

WEEK 4:

Q1: Prove Newton-Raphson method analytically
Q2: What are the various methods to obtain solutions of non-linear equations.
Q3: Solve the following system of linear equations using Gauss Elimination Method

2x1 + 8x2 + 2x3 = 14
x1 + 6x2 - x3 = 13
2x1 - x2 + 2x3 = 5

Q4: Give the programmatic implementation of Gauss elimination method
Q5: Solve the following system of linear equations using Gauss Jordon Method.

2x1 - 2x2 + 5x3 = 13
2x1 + 3x2 + 4 x3 = 20
3x1 - x2 + 3x3 = 10

WEEK 5:

Q1: Give the programmatic implementation of Gauss Jordon method
Q2: What is the difference between the Gauss-Jordon and Gauss-Elimination.
Q3: Give examples of various direct and iterative methods to obtain solutions of non-linear equation.
Q4: Solve the following system of linear equations using Gauss Seidel Method, correct to three decimal digits.

$$10x1 + x2 + 2x3 = 44$$
$$2x1 + 10x2 + x3 = 51$$
$$x1 + 2x2 + 10x3 = 61$$

Q5: Give the programmatic implementation of Gauss Seidel method

WEEK 6:

Q1: What is Pivoting. How is it useful.

Q2: Write an algorithm for implementing an inverse of a matrix in place with pivoting

Q3: Solve the following system of equations using LU Decomposition.

$$2_{x1}+ 8_{x2}+2_{x3}=14$$
$$_{x1}+ 6_{x2} - \; _{x3}=13$$
$$2_{x1}- \; _{x2}+2_{x3}=5$$

Q4: Programmatic implementation of LU Decomposition.

WEEK 7:

Q 1: Given the table of values as:

X	0	1	2	3
Y(x)	0	2	8	27

Find y (2.5) using Langrangian method.

Q2: Give the programmatic implementation of Langrangian method.

Q3: Given the table of values as:

X	2.0	2.25	2.50	2.75	3.0
Y(X)	9.00	10.06	11.25	12.56	14.00

Find y (2.35) using Newton’s forward difference interpolation formula.

Q4: Given the table of values as:

X	2.5	3.0	3.5	4.0	4.5
Y(X)	9.75	12.45	15.70	19.52	23.75

Find y (4.25) using Newton’s backward difference interpolation formula.

Q3: When is Newton’s forward interpolation method used

Q4: What do you mean by interpolation, inverse interpolation and extrapolation

Q5: When are iterative methods preferred over direct methods.

WEEK 8:

Q 1: What are the ways to approximate a function by a polynomial? Describe each in brief

Q2: we want to compute sin(x) correct to three significant digits. Obtain a series with minimum number of terms using Taylor series.

Q3: programmatic implementation of taylor series.

We want to compute sin(x) correct to three significant digits. Obtain a series with minimum number of terms using Chebyshev series.

Q4: programmatic implementation of Chebsyshev series.

WEEK 9:

Q1: Evaluate $\int_1^2 e^{\frac{-1}{2}x}dx$ using four intervals using the 1/3 Simpsons method.

Q2: Give the derivation of Simpsons method.

Q3: Evaluate $\int_1^2 e^{\frac{-1}{2}x}dx$ using four intervals using the Trapezoidal method.

Q4: Give the derivation of trapezoidal method.

Q5: Programmatic implementation of Eulers method.

WEEK 10:

Q1: Give the derivation of Euler's method
Q2: Given dy/dx = xy with y(1) =5. Find solution correct to decimal positions in the interval [1,1.5] using step size h=0.1 using Eulers method.
Q3: Write the "predictor" and "corrector formula" used in modified-Euler method
Q4: Given dy/dx = xy with y(1) =5. Find solution correct to decimal positions in the interval [1,1.5] using step size h=0.1using Modified Eulers method.
Q5: What are single-step and multiple-step methods. Give examples.

WEEK 11:

Q 1: Give a brief idea about runge-kutta (RK) methods.
Q2: Given dy/dx = xy with y(1) =5. Find solution correct to decimal positions in the interval [1,1.3] using RK second order method (step size h=0.1)
Q3: Provide the programmatic implementation of RK 2nd order method.
Q4: Given dy/dx = xy with y(1) =5. Find solution correct to decimal positions in the interval [1,1.3] using RK second order method (step size h=0.1)
Q5: Provide the programmatic implementation of RK 2nd order method.

WEEK 12:

Q 1: Explain the following terms with suitable examples.

i. Differential equation
ii. Solution of Differential equation
iii. Order of differential equation

Q2: Differentiate between the following:

i. Single step and multiple step methods
ii. Ordinary and partial derivates
iii. Ordinary and partial differential equations

Q3: Using Modified Euler's method, find the solution of the following differential equation dy/dx = $x+y^{2,}$ for x= 1.1,1.2,1.3. Given that y=1, x=1.

Subject Name: Numerical Techniques Lab

1. To deduce error involved in polynomial equation.
2. To Find out the root of the Algebraic and Transcendental equations using Bisection method.
3. To Find out the root of the Algebraic and Transcendental equations using Regula-Falsi method.
4. To Find out the root of the Algebraic and Transcendental equations using Newton-Raphson method.
5. To Find out the root of the Algebraic and Transcendental equations using Iterative method.
6. To implement Numerical Integration using Trapezoidal rule.
7. To implement Numerical Integration using Simpson 1/3 rule.
8. To implement Numerical Integration Simpson 3/8 rule.
9. To implement Newton's Forward Interpolation formula.
10. To implement Newton's Backward Interpolation formula.
11. To implement Gauss Forward Interpolation formula.
12. To implement Gauss Backward Interpolation formula.
13. To implement Bessel's Interpolation formula.
14. To implement Sterling's Interpolation formula.
15. To implement Newton's Divided Difference formula.
16. To implement Langrange's Interpolation formula.
17. To implement Numerical Differentiations.
18. To implement Least Square Method for curve fitting.

Subject Code: MCA20107DCE
Subject Name: Accounting and Management Control

Basic Accounting and conventions underlying preparation of Financial Statements (balance sheet highlighting accounting equation. profit and loss statement; accounting processes; basic accounts, trial balance and financial statements; issues such as provisions for bad debts tax, dividends, losses such as bad debts, missing information, classification effect, cost of assets, rentals, etc); Income Measurement (revenue; recognition and matching costs and revenues; inventory valuation); Depreciation Accounting; Intangible Assets Accounting; Understanding published annual accounts including funds flow statement.

Basic Cost Concepts: (introduction; cost classification; allocation, appointment and absorption; cost centers); Cost Analysis for Managerial Decisions (direct costing, break-even analysis; relevant costs; pricing; pricing-joint costs; make or buy; relevant fixed costs and sunk costs) Cost Analysis for Control (standard costing; variances; material, labour, overhead, sales, and profit); Standard Cost Accounting (budgeting and control; elements of budgeting; control of manufacturing and manufacturing expenses; performances appraisal, evaluation of cost control systems).

Introduction to Management Control Systems; Goals, Strategies, and Key Variables; Performance Measures; Responsibility Centers and Transfer Price; Investment Centers; Reporting Systems; Management by Objectives; Budgeting and Control; Organizational Relationships in Control; Control Dynamics; Top Management and Control; Strategic and Long-Range Planning; Control of Service Organizations; Control of Projects; Control of Non-Profit Organizations; Control of Multinational Companies.

References

Bhattacharya, S.K., and Dearden, John, "Accounting for Management", Prentice Hall of India, New Delhi.
Chadwick, "The Essence of Financial Accounting", Prentice Hall of India Pvt. Ltd., New Delhi.
Chadwick. "The Essence of Management Accounting", Prentice Hall of India Pvt. Ltd., New Delhi.
Homgren, Sundem and Selto (9th ed), "Introduction to Management Accounting", Prentice Hall of India Pvt. Ltd.
Welch, Hilton and Gordon (5th ed). "Budgeting: Profit Planning and Control", Prentice Hall of India Pvt. Ltd., New Delhi.

Subject Code: MCA20108DCE
Subject Name: Optimization Techniques

Unit I [12L]

Linear Programming Problem (LPP): Formulating LPPs, Simplex Algorithm; Duality in LPP: Duality Theorems **[4L]**

Transportation Problems: Mathematical Formulation of Transportation problem, Methods of selecting initial basic feasible solution: Matrix minima method, North-West Corner Rule, Vogel’s Approximation Method, Unbalanced Transportation Problem**[6L]**

Assignment problems: Algorithm, Unbalanced Assignment Problem, Hungarian Method**[2L]**

Unit II [12L]

Game theory: Definition and Terminologies; Pure Strategy: saddle point, Game with two saddle points; Mixed strategies: games without saddle points, 2Xn games, Dominance Property**[5L]**

Sequencing models: Sequencing of n jobs on two machines and three machines with no passing**.[2L]**

CPM- Determination of critical tasks. PERT- probability of completing the project on schedule**[5L]**

Reference Books:

1. H.A.TAHA,” Operations Research”. Pearson Education
2. S.D. Sharma,” Operations Research & Optimization”.
3. KantiSwaroop, “ Operations Research and Applications
4. R. PanneerSelvam :Operations Research”
5. N.D. Vohra “Quantitative Techniques”

General Elective Courses

Subject Code: MCA20109DCE
Subject Name: Matlab

Unit 1
Basics of MATLAB, Overview of features and workspace, Data types , Arrays : Initialization and definition, Array functions, 2---D Arrays, Multidimensional Arrays, Processing Array elements, Array sorting, **Matrices:** Matrix Operations, Matrix Functions, Manipulating matrices, Special Matrices. Decision Making using If---Else and Switch,

Unit 2
Function definitions, Function arguments, Function returns, Embedded Functions, Files and I/O, Reading from a file, Writing to a file, Formatting output, For Loops, Do While Loop, Plots and Graphs, Plot Types, Plot Formatting, Multiple Plots , Plot Fits : Extrapolation and Regression Solving basic matrix equations, modelling and solving system of equations.

Text Book :MATLAB Programming for Engineers by Stephen J. Chapman.

References:

1. Introduction to MATLAB for Engineers by William J. Palm III.
2. Essentials of MATLAB Programming by Stephen J. Chapman
3. MATLAB Guide to Finite Elements: An Iterative Approach by Peter I. Kattan.
4. An Introduction to Scientific Computing: Twelve Computational Projects Solved with MATLAB.
5. A Guide to MATLAB: For Beginers and Experienced Users by Brian R. Hunt (Editor).
6. Solving Odes with MATLAB by Lawerence F. Shampine.

Semester - II

Subject Code: MCA20201CR
Subject Name: Data Structures using C++

Unit I [12 L]

Data types/objects/structures, Data structures and its types, Representation and implementation.(**2L**) Linear Data Structures: Array representation, operations, applications and limitations of linear arrays, 2-dimensional arrays, matrices, common operations of matrices, special matrices, array representation of Sparse matrices[**4L**].

Linked Lists: Representation of Linear Linked List, Operations like creating , search an element inserting an element, deleting an element, reversing a list, merging two list, Deleting entire list. Linked list application, Polynomial Manipulation, Representing Sparse Matrices[**6L**]

Unit II [12 L]

Stack, Representation of stack in memory, Operations on Stacks, Implementation of Stack using arrays and linked list, Multiple Stacks: Representing two stacks and more than two stacks, Applications of stacks: Parenthesis Checker, Infix to postfix procedure, Evaluating expressions in postfix notation, Sparse Matrix Representation. Implementation of recursion using stack [**6L**]
Queues, Representation of Queue in Memory, Operations on Queue, Implementation of Queue using arrays and linked list, Circular Queue and its operations, Representation and implementation, Multiple Queues, DEQUE, Priority Queue, ,Linked Queue, Multiple Priority queue, Heap Representation of a Priority Queue, Applications of Queues.[**6L**]

Unit III [12 L]

Trees, Definitions, terminologies and properties ,Binary tree representation ,traversals and applications, Threaded binary trees, Binary Search Trees, AVL Trees, M-way Search Trees, B-trees, B*-trees[**8L**].
Graphs, Terminology, Graph representations, Traversal Techniques, Operations on Graphs, Applications of Graphs[**4L**]

Unit IV [12 L]

Minimum spanning trees, Shortest Path Algorithms in Graphs, Eulerian Tour, Hamiltonian Tour [**4L**]
Direct Address Tables, Hash Table, Different Hash functions, resolving collisions, rehashing, Heap Structures, Binomial Heaps, Leftist Heaps.[**4L**].
File Organizations: Sequential File Organization, Relative File Organization, Indexed Sequential File Organization, Multiple Key File Organizations: Inverted File and Multi-List Organizations[**4L**]

Text book:

1. SartajSahni, “Fundamentals of Data Structures in C++”, Galgotia Pub

References:

1 Heileman:data structure algorithims&Oop Tata McGraw Hill
2 Data Structures Using C – M.Radhakrishnan and V.Srinivasan, ISTE/EXCEL BOOKS
3 Weiss Mark Allen, “Algorithms, Data Structures, and Problem Solving with C++”, Addison Wesley.
4 Data Structures and Algorithms – O.G. Kakde& U.A. Deshpandey, ISTE/EXCEL BOOKS
5 Aho Alfred V., Hopperoft John E., UIlmanJeffreyD., “Data Structures and Algorithms”, Addison Wesley
6 Drozdek- Data Structures and Algorithms,Vikas
7 Tanenbaum A. S. , “Data Structures using ‘C’ ”

Subject Name: Data Structures in C++ Lab

Data Structures Lab Programming Assignments [One per week]

Week 1

Q1. Write a program in C++ to insert, delete and update the contents of an array.

Q2. Write a program in C++ to perform various operations on matrices.

Q3. Write a program to multiply two sparse matrices?

Q4. Write a program in C++ to implement different string manipulation operations?

Week 2.

Q1. Write a program to implement singly linked list?

Q2. Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a linked list.

Q3. Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a linked list.

Q4. Write a program in C++ to reverse a linked list by changing the link in the nodes?

Q5. Write a program to add two polynomials represented as linked list?

Week 3

Q1. Write a program in C++ to multiply two polynomials represented as linked lists?

Q2. Write a program in C++ to implement a doubly linked list?

Q3 Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linked list.

Q4. Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linked list.

Q5 Write a program to implement different operations of a circular linked list

Week 4

Q1. Write a program to implement various operations on an array based stack?

Q2. Write a program to implement various operations on an stack represented using linked list.

Q3. Write a program to demonstrate the use of stack in checking whether the arithmetic expression is properly parenthesized?

Q4. Write a program to demonstrate the use of stack in converting an arithmetic expression from infix to postfix?

Q5. Write a program to demonstrate the use of stack in evaluating an arithmetic expression in postfix notation?

Week 5

Q1. Write a program to demonstrate the use of stack in implementing quicksort algorithm to sort an array of integers in ascending order?

Q2. Write a program to demonstrate the implementation of various operations on a linear queue represented using a linear array.

Q3. Write a program to demonstrate the implementation of various operations on a Circular queue represented using a linear array.

Q4. Write a program to demonstrate the implementation of various operations on a queue represented using a linked list?

Week 6.

Q1. Write a program to demonstrate the use of multiple stacks?
Q2. Write a program in C++ to create a binary tree?
Q3. Write a program to implement the traversal techniques of a binary tree?
Q4. Write a program to C++ to create a binary tree?

Week 7.

Q1. Write a program to delete a node in a binary search tree?
Q2. Write a program to implement the different operations of an AVL tree?
Q2. Write a program to implement the different operations of a threaded binary tree.
Q2. Write a program to implement the different operations of a M-way search tree?

Week 8.

Q3. Write a program to implement the different operations of a B- tree?
Q4. Write a program in C++ to implement the different operations of a B+ tree?

Week 9

Q1. Write a program in C++ to implement the different operations of a B* tree?
Q2. Write a program in C++ to Multi-dimensional binary search trees

Week 10

Q1. Write a program in C++ to find the edges of a spanning tree using Prims Algorithm?
Q3. Write a program in C++ to implement the graph using different representations?
Q4. Write a C++ program to illustrate the traversal of a graph using Breadth First Search?
Q5. Write a C++ program to illustrate the traversal of a graph using Depth First Search?

Week 11

Q2. Write a program in C++ to find the shortest path in a graph using Warshalls Algorithm.
Q3. Write a C++ program to in C++ to find the shortest path in a graph using Modified Warshalls Algorithm.
Q4. Write a C++ program to in C++ to find the shortest path in a graph using Dijkstra’s Algorithm.
Q5. Write a C++ program in C++ to implement Euler Graphs?

Week 12

Q1. Write a program in C++ to implement Hamilton Graphs?
Q2. Write a program in C++ to implement Planner Graphs?
Q3. Write a program to C++ to implement Kruskals Algorithm?
Q4. Write a program to C++ to find the cycles in a graph?

Week 13

Q1. Write a C++ program to implement various hashing techniques?
Q2. Write a C++ program to demonstrate the concept of rehashing?
Q3. Write a C++ program to create Max and Min heaps?
Q4. Write a C++ program to create Binomial and Leftist heaps?

Subject Code: MCA20202CR
Subject Name: Software Engineering

UNIT I [12L]

Concept of Software and its characteristics, Concept of Software Engineering, Software Engineering Challenges, Software Engineering Approach, Software Engineering Process Models - Waterfall Model, Prototyping Model, Incremental and Iterative Models, Spiral Model, Concurrent Development Model, Agile Process Model, etc. (**6L**)

Software Process and Project Measurement: Measures, Metrics and Indicators, Size -Oriented Metrics vs. Function - Oriented Metrics. Capability Maturity Model Integration (CMMI), COCOMO Model.**(6L)**

UNIT II [12L]

Introduction to Requirements Engineering - Why, What and Where. Requirements Types: functional and non-functional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques. Requirement Analysis and Modelling, Requirements prioritization, verification, and validation.

UNIT III [12L]

Basics of Design Engineering**,** Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology. **(5L)**

Object-Oriented Design, OO Concepts, Design Concepts, Design Methodology, Object-oriented analysis and design modeling using Unified Modeling Language (UML), Dynamic & Functional Modeling, Design Verification. **(7L)**

UNIT IV [12L]

Software Testing – Basics of Software Testing, Testing Terminology, Adequacy Criteria, Static vs. Dynamic Testing, Black Box vs. White Box Testing. Structural testing and its techniques. Functional Testing and its techniques, Mutation testing and its techniques, Random Testing and its techniques. Non-Functional Testing like Reliability, Usability, Performance and Security Testing **(6L)**

Introduction to Software Reliability: Basic Concepts, Software Reliability metrics, Operational Profile, Reliability Estimation and Predication, Reliability and Testing , Clean room approach for building correctness, Concept of Software reengineering **(6L)**

Reference Books:

1. Software Engineering: Principles and practice, 3rd Edition, ***Hans Van Vliet,*** *Wiley.*
2. Pankaj Jalote - An Integrated approach to Software Engineering, 3rd edition, Narosa Publication.
3. Sommmerville - Software Engineering. Pearson , 7/e , 2006
4. Software reliability engineering 1stedition paperback, ***R. Subburaj***
5. James F. Peters Software Engineering – An Engineering Approach, Wiley& Sons
6. Guide to Advanced Software Testing (Computing), **Hass, Anne Mette Jonassen**

Subject Name: Software Engineering (Tutorial)

1. Study and compare different software process models and prepare comparison based analysis report on it. Also identify the suitable applications for the individual process model.

2. **Case Study:** You are the technical lead in your company for a new product aimed at university research scientists: *an electronic lab notebook (ELN).* The marketing team has given you the following information. The scientists who will use the ELN traditionally record their experiments in paper notebooks, but increasingly they are looking for digital solutions which allow them to share their information with other scientists they are working with, such as Wikis and Dropbox, which make frequent functionality changes. A small amount of market research indicates that the scientists vary considerably in what the capabilities they would like in an ELN, and the ELN solution will need to change over time as the scientists' needs change. However the scientists all say an ELN should be easy to use and quick to use. The scientists work in labs headed by a senior academic (the Lab Head). Provision of computing resources for the lab (hardware and software) varies between labs. Some have all their needs met by the central university IT department; others manage all their own hardware and software and depend on central IT only for network connections; most are somewhere in between.
    a. Which software development process is likely to be most appropriate for developing the software system?
    b. Explain why, contrasting your choice with one other software development process.
3. You are in charge of a large team based in New Delhi, Mumbai, Chennai and Kolkata, developing the train operations control system for the new HS2 high speed train.
    a. Would you adopt XP? If not, then what process would you adopt and why?

4. Develop a web-based application to calculate Function Points. The application should give user option to select available values for weighing factors and complexity adjustment factors. The application should also calculate various metrics like productivity, quality, cost etc.

5. Calculate the effort, duration and average persons required using the above details for basic CoCoMo model. Given
    Number of user inputs=15
    Number of user outputs=14
    Number of user inquiries=8
    Number of files =6
    Number of external interfaces=13
    The weighing factors have following values (inputs=3, outputs=5, inquiries=3, files=15, interfaces=7). Assume that all complexity adjustment values are average and 1 function point = 25 LOC. The Values of Constant used in basic CoCoMo model. a=2.4, b=1.05, c=2.5, d=0.38.

6. Choose any standard application (e.g. Library Management System, Railway Reservation system, Hospital management System, University Admission system).
    a. Identify the different requirements of the application.
    b. To identify the various elicitation techniques and their usage for the identified requirements.
    c. Prepare a requirement document for the same as per the IEEE standard.
    d. To classify the requirement into functional and non-functional requirements.
    e. Use various validation techniques to validate your requirement document.

7. Design the system using structured as per the instructions and applicability.
    i. Implement the design using DFD, ER diagrams and structure chart whichever applicable.
    ii. Identify various processes, data store, input, output etc. of the system.
    iii. Use processes at various levels to draw the DFDs.
    iv. Identify various modules, input, output etc. of the system and ask students to analyze.
    v. Use various modules to draw structured charts.

8. Design the system using object oriented design as per the instructions and applicability. Use UML modelling technique appropriately.

i. Identify various processes, use-cases, actors etc. of the system and ask students to analyze. Use processes at various levels to draw the use-case diagram.
ii. Identify various elements such as classes, member variables, member functions etc. of the class diagram. Draw the class diagram.
iii. Identify various elements such as various objects of the object diagram. Draw the object diagram.
iv. Identify various elements states and their different transition of the state-chart diagram. Draw the state-chart diagram.
v. Identify various elements such as controller class, objects, boundaries, messages etc. of the sequence diagram. Draw the sequence diagram as per the norms.
vi. Identify various elements such as for the sequence diagram of the collaboration diagram. Draw the collaboration diagram as per the norms
vii. Identify various elements such as different activity their boundaries etc. of the activity diagram. Draw the activity diagram.
viii. Identify various elements of the component diagram such as the various components like client, server, network elements etc. Draw the component diagram.
ix. Identify various elements such as the hardware components of the deployment diagram. Draw the deployment diagram.

9. Identify the appropriate and necessary testing approach and design detailed test cases according to IEEE format for test cases.

a. Identify various modules of the system so that they can be tested stand-alone.
b. Identify the groups of the module that can be tested together in integration.

10. Write a program to calculate reliability of any software products. The user will input necessary details, the program should compute the reliability and display appropriate reliability metrics.

Subject Code: MCA20203CR
Subject Name: Artificial Intelligence

Unit I

Introduction and historical perspective, AI – disciplines and applications, Theories of Intelligence, Detecting and Measuring Intelligence, Turing Test. Industry Successes **4 Lectures**

Knowledge Representation: Propositional Logic, First Order Logic, Inference in First Order Logic, Prepositional Versus First Order Logic **4 Lectures**

Expert Systems, Forward chaining, backward chaining, Conflict Resolution, Knowledge representation, Representing Uncertain Knowledge and Reasoning with Uncertain Information
4 Lectures

Unit II

Fuzzy Logic, Fuzzification, Fuzzy Sets, Operations on Fuzzy Sets, Hedges, Reasoning in Fuzzy Logic-Mamdani Inference **4 Lectures**

Search Algorithms –Soundness & Completeness, Local search algorithms (gradient ascent, simulated annealing, genetic algorithms) **4 Lectures**

Heuristic search, A* Algorithm, ID A* algorithm, Genetic Algorithms-operations, applications, limitations
4 Lectures

Unit III

Machine Learning, Types of Learning, Knowledge in Learning, Inductive Learning Algorithms, Learning Decision Trees, ID3 Algorithm **4 Lectures**

Supervised Learning, Support Vector Machines, Using SVM for linearly inseparable data, SVM for linearly inseparable data, Statistical Learning Methods **4 Lectures**

Unsupervised Learning, K-Means Algorithm, Classification and clustering, Classification based on distance-Euclidean & Mahalanobis, Reinforcement Learning **4 Lectures**

Unit IV

Neural Computing, Network Architectures and Learning Algorithms, Perceptron and its Limitations, Activation Functions **4 Lectures**

Multilayer Neural Networks, Training by Error Back Propagation, SelfOrganising Nets, Kohonen Self-Organising Net **4 Lectures**

Generalization, Overtraining, Applications of Neural Nets, Convolutional Nets, Deep Learning

4 Lectures

Artificial Intelligence Tutorials

1. Given a real life situation
    - identify the characteristics of the environment
    - identify the percepts available to the agent
    - identify the actions that the agent can execute
    - suggest the performance measures to evaluate the agent
    - recommend the architecture of the desired intelligent agent
2. Undertake a comparative study on popular programming languages used in AI.
3. Translate commonly used English sentences into their equivalent logical expressions.
4. Given a problem description, formulate it in terms of a state space search problem. Analyze the given problem and identify the most suitable search strategy for the problem. Make an analysis of the properties of proposed algorithms in terms of
    - time complexity
    - space complexity
    - optimality
5. Represent a real life problem in terms of fuzzy rules and apply inferencing mechanism on this knowledge-base
6. Compare Bayesian reasoning and certainty factors. Which applications are most suitable for Bayesian reasoning and which for certainty factors? Why? What is a common problem in both methods?
7. Techniques and tools for determine the final output of a fuzzy system mathematically and graphically.
8. Using Weka (open source software) on any dataset, perform data pre-processing, classification, regression, clustering, association rules, and visualization.
9. Features of Matlab Fuzzy Logic Toolbox
10. Using Simulink block for analyzing, designing, and simulating systems based on fuzzy logic.
11. Solve linear, quadratic, integer, and nonlinear optimization problems using Matlab's Optimization Toolbox
12. Using Matlab Neural Network Toolbox, design, implement, visualize, and simulate neural networks.

Subject Name: Artificial Intelligence Lab

Week 1: Using Fuzzy Logic solve the following Tipping problem in Matlab:
Given two sets of numbers between 0 and 5 (where 0 is for very poor, and 5 for excellent) that respectively represent quality of service and quality of food at restaurant, what should tip be?
Week 2: Solve 2-input 1-output project risk prediction problem using Mamdani Inference. Make necessary assumptions.

Week 3: Solve a 2-variable optimization problem by formulating it as a Genetic Algorithm problem.

Week 4: Optimize simple 2-variable functions using Simulated Annealing

Week 5: Implement A* algorithm for 8-Puzzle problem

Week 6: Implement A* algorithm Missionaries and Cannibals problem

Week 7: Create a decision tree from given dataset using ID3 algorithm

Week 8: Given a dataset which is linearly separable, classify the datapoints using SVM.

Week 9: Given a 2-class dataset which is linearly inseparable, demonstrate what needs to be done so that SVMcan be used to classify it.

Week 10: Cluster a given 2-dimensional dataset using k-means algorithm.

Week 11: Demonstrate Neural Networks using different activation functions

Week 12: Implement Back-propagation Algorithm in Matlab.

Subject Code: MCA20204CR
Subject Name: Computer Graphics & Multimedia

Unit I

Introduction to Computer Graphics. Applications of Computer Graphics. Graphic Display Devices: Refresh Cathode Ray Tubes, Raster-scan Displays, Random-Scan displays, Color CRT Monitors, Concept of Double Buffering, Lookup tables (6L)

2-D Graphics: Cartesian and Homogeneous Coordinate Systems. Line drawing algorithms (Bressenham's and DDA). Circle and Ellipse Drawing Algorithms. (6L)

Unit II

2-Dimensional Transformations. Concepts of Window & Viewport, Window to Viewport Transformations, Normalization transformation. (3L)

Composite Transformations: General pivot point rotation, General fixed point scaling, reflection w.r.t line $y = x$, reflection w.r.t line $y = -x$(5L)

Transformation between coordinate systems, Affine transformations, Raster methods for transformations, (4L)

Unit III:

Filling, Boundary and Floodfill algorithms(2L)

Clipping, Line Clipping Algorithms (Cohen-Sutherland Algorithm), 3-D Graphics, Projections: perspective and parallel projection transformations. (6L)

3-Dimensional Transformations. Hidden Surface Removal Techniques, Z-Buffer Algorithm, Back Face

Detection. (4L)

Unit IV

Curves and Surfaces: Spline specification, Interpolated& Approximated Splines. Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. (3L)

Introduction to Multimedia, Visual elements: BMP file format, PCX file format, WAV and MP3 file format; Sound elements; Multimedia Storage- Compact disk(CD) and Digital Versatile Disk(DVD)(9L)

Text Book : Hearn and Baker " Computer Graphics" 2nd Edition , Pearson Education.

Reference Books

1. W.M.Newman and Sproull. "Principles of interactive Computer Graphics" ,TMH
2.Steven Harrington." Computer Graphics a Programming Approach" McGraw Hill.
3. Plastock and Kelley. "Schaums outline of theory and problems of computer Graphics"
4. David F Frogers and J Alan Adams. "Procedural Elements of Computer Graphics" McGraw Hill
5. David F Rogers and J Alan Adams. "Mathematical Elements of Computer Graphics" McGraw Hill
6. James. D. Foley, A Van dam etal "Computer Graphics" Pearson.
7. Sinha and Udai , "Computer graphics", TMH

Discipline Centric Elective Courses

Subject Code: MCA20205DCE
Subject Name: Web Programming

Unit - I

Adobe Photoshop Environment, Interface tour of Photoshop and Palettes, Color Modes and Resolutions, Using different Photoshop tools. **[3L]**

Working with Layers Grouping and Smart objects, Image Adjustments, Layer Masking and Layer Clipping,Using Blending Options, Filters, Photoshop actions, Animation tools**[3L]**

Markup Language, Basic Structure of HTML , Meta Tags, Document Structure Tags, Formatting Tags, Text Level formatting, Block Level formatting, List Tags, Hyperlink tags, Image and Image maps, Table tags, Form Tags, Frame Tags, Executable content tags, Tables as a design tool, Forms, Creating Forms**.[4L]**

XHTML, Difference between HTML and XHTML, diff Doc Typtes, HTML5, New in HTML5, New Structure Tags and Form tags in HTML5**.[2L]**

Unit II

Style Sheets : Different approaches to style sheets, Using Multiple approaches, Linking to style information in s separate file, Setting up style information. [**2L**]

Java Script :, JavaScript Objects, JavaScript Security, Operators : Assignment Operators, Comparison Operators, Arithmetic Operators, Logical Operators, String Operators, Special Operators, ? (Conditional operator), ,(Comma operator), delete, new, this, void Statements : Break, comment, continue, delete, do … while, export, for, for…in, function, if…else, import, labelled, return, switch, var, while, with, Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp Document and its associated objects : document, Link, Area, Anchor, Image, Applet, Layer Events and Event Handlers : General Information about Events, Defining Event Handlers and events.[**6L**]

jQuery UI, How jQuery UI Works, Theming jQuery UI, Using jQuery UI Theme Roller, Widget Factory, Theming jQuery UI, using jquery core, events and effects**.[4L]**

Unit III

PHP , Server-side web scripting, Installing PHP, Adding PHP to How PHP scripts work , Basic PHP syntax , PHP data types, PHP Variables, Operators in PHP, Conditional Statements , Loops (If, If else and Switch) [**4L**]

Strings, Arrays and Array Functions, Numbers, PHP Function: User-Defined Functions, Inbuilt functions, Basic PHP errors / problems, Working with Forms, Designing a Form, $_GET and $_POST, HTML and PHP code, User Input, Form Validation, Cookies, File uploading, Sessions **[5L]**

Object oriented concepts, Define a class, Class attributes, Creating an Object,Object properties, Object Methods, Object constructors and destructors, Static Method, Class Inheritance, Abstract Class**[3L]**

Unit IV

Advanced PHP and MySQL : How to PHP MySQL Integration Works? ,Creating a database connection, Selecting the DB, Basics of SQL, SQL Syntax, CRUD Operations, Inserting data in database, Inserting data with a File, Retrieving data from Database, Retrieving data with specific criteria, Updating records, Searching the records, Alter table structure, Deleting the records Dropping tables. Emailing with PhP.

References :

1. Web Design The complete Reference, Thomas Powell, Tata McGrawHill
2. HTML and XHTML The complete Reference, Thomas Powell, Tata McGrawHill
3. JavaScript 2.0 : The Complete Reference, Second Edition by Thomas Powell and Fritz Schneider
4. PHP : The Complete Reference By Steven Holzner, Tata McGrawHill

Subject Name: Web Programming Lab

Week 1:

1. Open any picture and make use of rectangular and elliptical selection tools to select portions of the image and paste it in another image. Also make use of move tools.
2. Make use of the Lasso- and Polygonal Lasso Selection Tools, Copy, Paste Into, Move Tool, Zoom Tool, Quick Select Tool (or Magic Wand Tool), Invert Selection, Copy, Paste Transform tools for editing an image.
3. Edit any image using the following tools, Paint Bucket Tool, Color Picker, Brush Tool.
4. Select an image and make use of Text Tool, Selection Tools, Copy, Paste, Transform, Move Tool, Opacity, Eraser Tool to perform different operations
5. Select any image of your choice and make use of the Brush Tool, Smudge Tool, Dodge Tool, Burn Tool, Layer Styles, Modes, The Shape Tools, the Styles palette.

Week 2

1. Applying different filters on a image and make use of different layers.
2. Create a page banner from scratch using browser-safe colors
3. Make the illusion of an image fitting inside your text using clipping mask.
4. Create a Photoshop project which makes text have a reflecting effect.
5. Create a web page in Photoshop following the following 4 steps:-
   a. Creating a unique website header
   b. Designing your website layout in Photoshop
   c. Slicing your layout in Photoshop
   d. Creating your web site in Dreamweaver

Week 3

1. Create a html page with demonstrates the use of formatting tags image tags and other basic tags.
2. Create the different types of list in html
3. Create a table with the relevant tags and attributes
4. Create a html form in the table layout covering major form elements
5. Create a html page with links in one frame and the linked page opens in the other frame.
6. Make use of tables as the layout of the web page.

Week 4

1. Link an external style sheet with styles for basic tags.
2. Create a Navigation bar (with dropdown) with CSS, CSS Grid, CSS based sticky footer.
3. Create a CSS3 based Zebra striped table, CSS3 based button, CSS3 image rounded shape
4. Create a stitched look, CSS3 Corner Ribbon, CSS3 blurry text effect, CSS3 speech bubble shape
5. Create image cross fade with CSS3 transition, Google Font, Font Awesome Vector Icons with Bootstrap3, Calculating Values With CSS3 calc().

Week 5

1. Write a JavaScript program to sum the multiples of 3 and 5 under 1000 ?
2. Write a JavaScript program to find the GCD of two numbers?
3. Write a JavaScript function to test whether a string ends with a specified string
4. Write a JavaScript function that returns the longest palindrome in a given string.
5. Write a program using Java Script that checks if two matrices have identical values in all the elements
6. Write a JavaScript program to check a credit card number and validate an email address?

Week 6

1.
    a. Change button text using jQuery
    b. Set background-image using jQuery CSS property
    c. Remove a specific table row using jQuery.
    d. Set value in input text using jQuery.
    e. Set a value in a span using jQuery.
    f. Find the class of the clicked element.
    g. Set href attribute at runtime using jQuery.
    h. Remove disabled attribute using jQuery.

2. Write the jQuery code to Toggle a specified class when an element is clicked?
3. Write a jQuery Code to get a single element from a selection?
4. Using jQuery Display the tag's name on click?
5. Using jQuery Attach a function to the focus event. The focus event occurs (display a message regarding the text field) when the <input> field gets focus?
6. Using jQuery Display the keyboard key which was pressed in a textbox?
7. Set background color of an element when the element (or any elements inside it) gets focus or loses focus.

Week 7

1. Create a simple HTML form and accept the user name and display the name through PHP echo statement
2. Write a PHP program to check the bits of the two given positions of a number are same or not.
3. Write a PHP program to remove duplicates from a sorted list.
4. Write a PHP program to compute the sum of the prime numbers less than 100
5. Write a PHP program to print out the sum of pairs of numbers of a given sorted array of positive integers which is equal to a given number?
6. Write a PHP script to remove all white spaces in an array.
7. Write a PHP script to create a two-dimensional array (4x4), initialized to 10. Write a PHP function to compares two multidimensional arrays and returns the difference.
8. Write a PHP script to combine (using one array for keys and another for its values) the following two arrays.

Week 8

1. Write a program to calculate and print the factorial of a number using a for loop.
2. Write a PHP script using nested for loop that creates a chess board?
3. Write a program that inputs a number from the user and display all armstong numbers upto the number entered using loops?

4. Write a function to reverse a string.
5. Write a function to sort an array.

Week 9

1. Write a simple PHP class which displays an introductory message like "Hello All, I am Scott", where "Scott" is an argument value of the method within the class.

2. Write a PHP class that calculates the factorial of an integer.

3. Write a PHP class that sorts an ordered integer array with the help of sort() function.
   Sample array : array(11, -2, 4, 35, 0, 8, -9)
   Output : Array ([0] => -9 [1] => -2 [2] => 0 [3] => 4 [4] => 8 [5] => 11 [6] => 35)

4. Calculate the difference between two dates using PHP OOP approach. Sample Dates : 1981-11-03, 2013-09-04
   Expected Result : Difference : 31 years, 10 months, 1 days

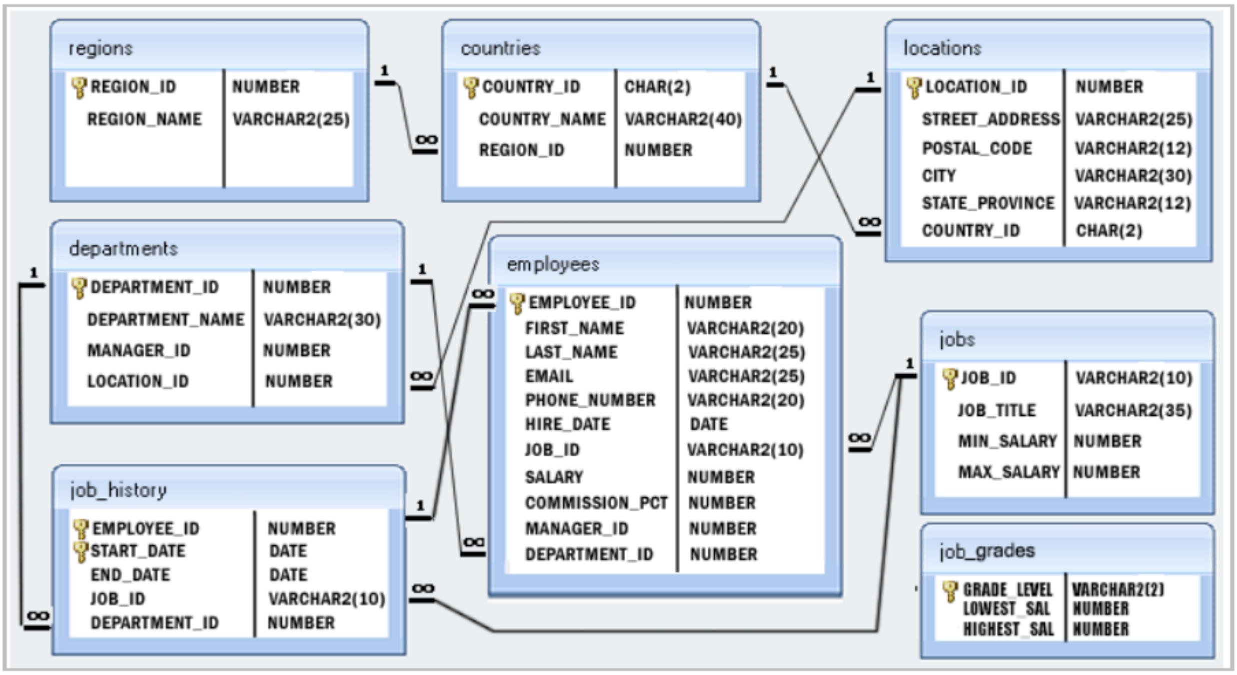
5. Write a PHP Calculator class which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request. For example :
$mycalc = new MyCalculator(12, 6);
echo $mycalc- > add(); // Displays 18
echo $mycalc- > multiply(); // Displays 72

6. Write a PHP script to convert a string to Date and DateTime.
Sample Date : '12-08-2004'
Expected Output : 2004-12-08
Note : PHP considers '/' to mean m/d/Y format and '-' to mean d-m-Y format.

Week 10

1. Write a PHP script to get time difference in days and years, months, days, hours, minutes, seconds between two dates
2. Write a PHP script to change month number to month name.
3. Write a PHP script to get yesterday's date.
4. Write a PHP script to get the current date/time of 'Australia/Melbourne'.
5. Write a PHP script to check if a date is a weekend or not.
6. Write a PHP script to add/subtract the number of days from a particular date
7. Sample Output : Original date : 2011-01-01
Before 40 days : 2010-11-22
After 40 days : 2011-02-10
8. Write a PHP function to get start and end date of a week (by week number) of a particular year
9. Write a PHP script to generate random 11 characters string of letters and numbers
10. Write a PHP function to create a human-readable random string for a captcha.

Week 11



1. Write the mysql code to create the database represented by following E-R diagram . Keep all the referential integrity constraints into consideration?
2. Insert the dummy data inside the tables making any assumptions as required if any ?

3. Write a SQL statement to insert records into the table countries to ensure that the country_id column will not contain any duplicate data and this will be automatically incremented and the column country_name will be filled up by 'N/A' if no value assigned for that column.
4. Write a SQL statement to insert rows in the job_history table in which one column job_id is containing those values which are exists in job_id column of jobs table.
5. Write a SQL statement to insert rows into the table employees in which a set of columns department_id and manager_id contains a unique value and that combined values must have exists into the table departments.
6. Write a SQL statement to insert rows into the table employees in which a set of columns department_id and job_id contains the values which must have exists into the table departments and jobs.

Week 12

1. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000.
2. Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order.
3. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 and are in department 30 or 100.
4. Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987.
5. Write a query to display the first_name of all employees who have both "b" and "c" in their first name.
6. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary is not equal to $4,500, $10,000, or $15,000
7. Write a query to get the total salaries payable to employees.
8. Write a query to get the minimum salary from employees table.
9. Write a query to get the maximum salary of an employee working as a Programmer.
10. Write a query to get the average salary and number of employees working the department 90.
11. Write a query to find the name (first_name, last_name) and hire date of the employees who was hired after 'Jones'.
12. Write a query to get the department name and number of employees in the department
13. Write a query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.
14. Write a query to display the department ID and name and first name of manager.
15. Write a query to display the department name, manager name, and city.
16. Write a query to display the job title and average salary of employees.
17. Write a query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job
18. Write a query to get the DATE value from a given day (number in N).
19. Write a query to get the firstname, lastname who joined in the month of June.
20. Write a query to get the years in which more than 10 employees joined.
21. Write a query to get first name of employees who joined in 1987.
22. Write a query to get department name, manager name, and salary of the manager for all managers whose experience is more than 5 years.
23. Write a query to get employee ID, last name, and date of first salary of the employees.
24. Write a query to get first name, hire date and experience of the employees
25. Write a query to get the department ID, year, and number of employees joined.
26. Write a query to update the portion of the phone_number in the employees table, within the phone number the substring '124' will be replaced by '999'.
27. Write a query to get the details of the employees where the length of the first name greater than or equal to 8.
28. Write a query to display the first word from those job titles which contains more than one words
29. Write a query to display the first eight characters of the employees' first names and indicates the amounts of their salaries with '$' sign. Each '$' sign signifies a thousand dollars. Sort the data in descending order of salary.

30. Write a query to display the employees with their code, first name, last name and hire date who hired either on seventh day of any month or seventh month in any year

Week 13.

1. Create a phpmysql connection which connects to the hr database using Phpobjects ?
2. Create a form to add using sign in and sign out , update and delete employee to the hr database?
3. Create a login , logout for every employee and list all the employee in the database?
4. Write a php script which emails the login details to the new employee along with his salary details fetch from the hr database?
5. Write a php script which demonstrates the use of sessions, cookies and data security which inserting in the database?

Subject Code: MCA18303CR
Subject Name: Web Programming Tutorial

Week 1
Q1. Show how to create a fiery text effect using filters, layer styles?
Q2. How do we design hover button and a banner for a webpage in Photoshop?
Q3. How is image slicing done?
Week 2
Q4. How do add a menu to a webpage?
Q5. How do we upload an image from a webpage?
Q6. Create a email registration form in html
Week 3
Q6. What are layouts and how do we create layouts using tables and css for a web ?
Q7. What are templates and how do we design templates?
Week 4
Q8. How do we link styles to our webpage internally as well as externally?
Q9. Write the CSS for various tags and use that for design the layout of a web page?
Week 5
Q1. Write the JavaScript code to validate a email registration form ?
Q2. Write the JavaScript code to generate dates between two dates?
Week 6
Q1. Create a form in jQuery which can accepts details of an employee like full name, parentage, address, date of birth , resume and photograph.
Week 7
Q1. Write a PHP script to remove nonnumeric characters except comma and dot. *Sample string :* '$123,34.00A'
Expected Output : 12,334.00
Q2. Write a PHP script to display time in a specified timezone?
Q3. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birthday.

Q4. Create two separate files. The first will contain a form with one input field asking for the user's favorite city. Use the post method for the form. Although this file contains no PHP code, on my localhost, it needs the .php extension to successfully call the second file.

The second file will contain PHP code to process the user's response. (In this case, something very simple.) After the user clicks the submit button, echo back Your favorite city is $city., where $city is the input from the form.
Week 8
Q1. Write a PHP script to find the maximum and minimum marks from an array of numbers?
Q2. Write a PHP script to generate random 11 characters string of letters and numbers.

Q3. Write a PHP script to convert scientific notation to an int and a float.

Sample scientific notation : 4.5e3

Expected Output : 4 & 4500

Q4. Write a PHP script to extract mantissa and exponent from a floating point value

Week 9

Q1. Create a php class to represent a person, and class student and class teacher to demonstrate inheritance?

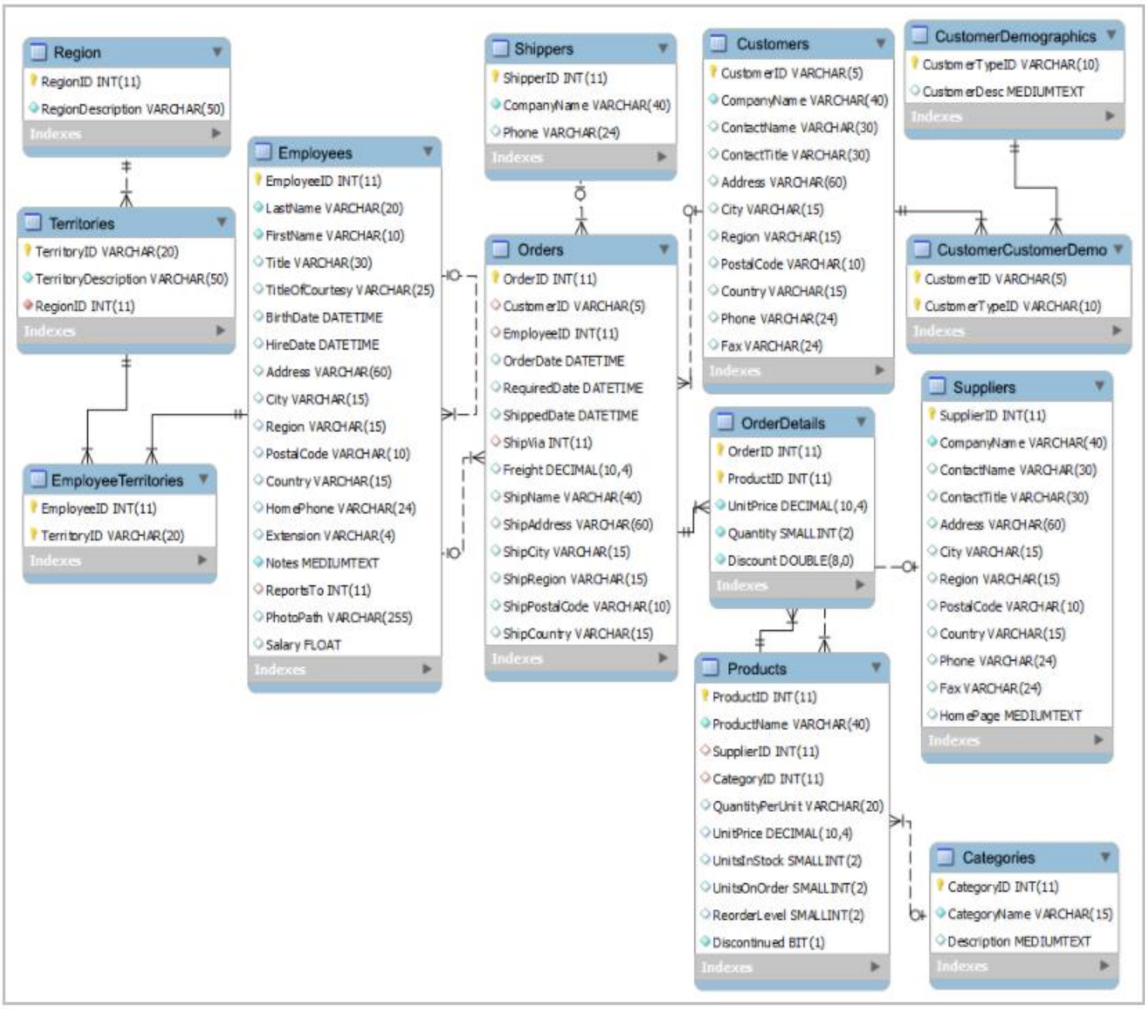
Q2. Calculate the difference between two dates using PHP OOP approach.
Sample Dates : 1981-11-03, 2013-09-04
Expected Result : Difference : 31 years, 10 months, 1 days

Week 10

Q1. Write the mysql code for the following database

Q2. Create a user with admin privileges in mysql to use for the connection with the northwind database?

Structure of 'northwind' database:



Week 11.

1. Insert the relevant test data inside the database?

Week 12.

1. Create the php form and required webpages for the above database for all the operations including insert, update, delete?

Week 13.

1. You have just been hired as the web programmer for a new startup company called, WebPro. WebPro wants to sell books about web programming on the web (makes sense, doesn't it). They have asked you to design their web site using PHP and MySQL. They want the following abilities:

   - browse inventory by category
   - search inventory by author and title
   - purchase multiple books; this includes the ability to
     - add books to shopping cart
     - view books in shopping cart
     - delete books in shopping cart
     - purchase books with credit card (only XCard is being accepted at this time)

   Note: Because WebPro is a startup, they have a limited inventory.

   You have decided to build the WebPro website incrementally.

Subject Code: MCA20206DCE
Subject Name: Cryptography and Network Security

Unit I: Need for Security-OSI Security Architecture, Security Attack – Threats, Vulnerabilities, and Controls, Security services-Confidentiality, Integrity, Authentication, and Availability. Types of Attacks.
Introduction to Number theory, Primality and Fermat's little theorem, Modular Arithmetic, Eulers theorem and Eulers PHI function, Euclidean and Extended Euclidean Algorithm

Cryptology, cryptography and cryptanalysis, Classical Encryption techniques, Substitution Techniques: Caesar and Vigenere cipher, Hill cipher and Playfair cipher, **Transpositions Techniques: Rail fence, Columnar Transition and OTP.**

UNIT II
Stream ciphers, Random Number generation, linear feedback shift registers, RC4 Algorithm,
Block Cipher and modes of operation, Data Encryption Standard (DES), Triple DES, Advance Encryption Standard (AES), Blowfish
Introduction to Public key cryptography: RSA, Multiple encryption and Brute force attack

UNIT III
Key management, Diffie Hellman Key exchange, Digital signatures,
Message Authentication codes MACs, Hash functions, Message Digest 5, Secure Hash Algorithm HMAC, Digital certificates, Email security, PGP protocol, S/MIME

UNIT IV
Overview of IP Security (IPSec) and security associations, Authentication Header (AH), Encapsulating Security Protocol (ESP)
Web security: Secure Socket Layer and TLS, Attacks and countermeasures
Intrusion detection, Firewalls and its types, Firewall configurations, malicious software's

References:

Stallings, William. *Cryptography and network security, 4/E*. Pearson Education India, 2006.
Paar, Christof, and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.
Stallings, William. *Network Security Essentials: Applications and Standards, 4/e*. Pearson Education India, 2003.

Subject Code: MCA20207DCE
Subject Name: Linux Programming

Unit I

Introduction to Linux: History of Unix and Linux. Linux basic features and applications. Linux Distributions. Linux Directory Structure. Linux Installation Procedure. [4L]

Introduction to Linux Shell: Shell - Definition and its types. Shell commands – Simple, Compound, Complex, and Separators. [2L]

General Commands: File and Directory manipulations commands. Manipulating File attributes. Process control commands. Commands to evaluate expressions. [4L]

Text Processing Utilities: head, tail, cut, paste, sort, uniq, tr, grep. Regular Expressions. [3L]

Unit II

Introduction to Shell Scripting: Shell Script – Structure and execution. Variables **-** Defining and accessing Local variables, Shell variables, and Environment variables. Scalar and Vector variables. [3L]

Input/Output: Input from user, Input redirection and Pipes. Output to terminal and Output redirection. [2L]

Flow Control and Loops: Expressions – Arithmetic, Relational and Logical. Flow control – if statement and test command, case statement and patterns. Looping – while loop, until loop, for loop and select loop. Loop control – break and continue. [3L]

Substitution and Quoting: Filename, Variable, Command and Arithmetic substitution. Quoting with Backlashes, Single Quotes and Double Quotes. Quoting Rules and Situations. [3L]

Functions and parameters: Creating and calling a function. Passing and retrieving parameters. Returning a value. Positional parameters. [2L]

Unit III

Introduction to X-Window System: History, features and components of X-Window System. Architecture and working of X-Window System (X-Server, X-Protocol, X-Client, and X-lib).

X-Window System limitations and pitfalls. [4L]

Desktop Environment: Window Manager, Client Application, and GUI Toolkit. Approaches for Cross-platform GUI development. [3L]

GUI Programming using Qt: Basic structure of a Qt program. Compilation tools – qmake, make and moc. Introduction to Signals and Slots. [2L]

Introduction to basic Qt Widget Classes:QLabel, QLineEdit, QPushButton, QCheckBox, QRadioButton, QSpinBox, QListBox, QComboBox (Common Properties, Methods and Slots). Layout Managers.QWidget Class and extending Widgets.Adding Custom Slots to Widgets. [4L]

Text Book: S. Veeraraghavan, *Teach YourselfShell Programming in 24 hours*, SAMS/Techmedia, 2007.

Reference Books:

1. C. Albing, J. Vossen, and C. Newham, *bash Cookbook*, O’Reilly Media, 2007.
2. T. Pavlidis, *Fundamentals of X Programming: Graphical User Interfaces and Beyond*, Kluwer Academic Publishers, 2002.
3. M. Dalheimer, *Programming with Qt: Writing Portable GUI Applications on Unix and Win32*, 2nd Edition, OReilly, 2002.
4. J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*, Prentice Hall, 2008.

Subject Name: Linux Programming (Lab)

List of experiments/practical-assignments for the course:

Week 1. This week students will learn how to acquire and install Linux operating system.

a. Download latest version of Linux operating system, preferably Fedora Core 20 (FC20) or above (Please visit https://getfedora.org/). Also, download latest version of Virtual Box for your preferred platform (Please visit https://www.virtualbox.org/).

b. Install FC20 on bare-machine after preparing it for single/dual boot configuration, or use Virtual Box to host FC20 in your preferred platform as instructed in the lecture.

Week 2. This week students will learn how to use basic file and directory manipulation commands.

a. Enter a series of commands on Linux shell to create an empty file, rename it, move it to another directory, add contents to it using *vi* editor, display its contents, and delete it.

b. Enter a series of commands on Linux shell to create a directory, rename it, add 7 empty files and directories to it, list its contents, move it to another directory, and delete it.

Week 3. This week students will learn how to manipulate permissions and evaluate expressions.

a. Create an empty text file, and use *chmod* to change its permission attributes by adding execute permissions for all users, removing write permission for group and others, and allowing read permission for only the user.

b. Enter a series of commands to calculate area of circle for radius 3 units.

Week 4. This week students will learn how to use text-processing utilities.

a. Enter a series of commands to process a text file to list its top 3 lines, bottom 3 lines, and 3rd word of every line.

b. Enter a regular expression in *grep* to list all lines of a text file that begin with a capital letter.

c. Enter a regular expression in *grep* to list all lines of a text file that contain an opening and closing parenthesis, with only letters and single-spaces in between.

Week 5. This week students will learn how to use flow control statements.

a. Write a shell script that reads a number from keyboard and checks whether the number is even or odd.

b. Write a shell script that takes backup of only those files residing in current directory which fulfill the following criteria:

i. are text files,

ii. have extension '.txt', and

iii. have size less than 3000 bytes.

Week 6. This week students will learn how to use looping statements.

a. Write a shell script that reads and displays a text file line-by-line while adding an extra line-space while displaying.

b. Write a shell script that iteratively calculates the factorial of a number that is read from the keyboard.

Week 7. This week students will learn how to use nested loops.

a. Write a shell script that uses a nested loop to find prime numbers from 2 to 100.

b. Write a shell script that declares and initializes an integer array of 10 elements and sorts its elements in ascending order using *bubble sort*.

Week 8. This week students will learn how to define a function, return value, and call it recursively.

a. Write a shell script that defines a function that takes two integers as parameters, calculates their sum, and returns the result to the caller.

b. Write a shell script that recursively calculates the factorial of a number (that is read from the keyboard) using functions.

Week 9. This week students will learn how to display basic Qt widgets, lay them on the canvas, and retrieve their values.

a. Write a C++ program using Qt classes to create an application that allows a user to enter Name (using QLineEdit), Age (using QSpinBox), Gender (using QRadioButton), Country of residence (QComboBox), Course preferences (using QCheckBox), and so on. The application should display the formatted information using QLabel after pressing the Submit button.

Week 10. This week students will learn how to use default Slots and create custom Slots.

a. Write a C++ program using Qt classes to create an application in which the text entered in a Textbox (QLineEdit) can be copied into another Textbox or Listbox (QListBox) or Combo-box (QComboBox) depending upon the push-button (QPushButton) pressed.

b. Write a C++ program using Qt classes to create an application in which selecting an item from a Listbox (QListBox) removes it from the Listbox and adds the same item to Combo-box (QComboBox).

Week 11. This week students will be tested to apply the knowledge gained in previous weeks.

a. Write a C++ program using Qt classes to create a 2-player tic-tac-toe game using QPushButton class.

b. Further modify the program above to include Computer as one of the players.

Week 12. This week students will learn how to validate the user-input.

a. Write a C++ program using Qt classes to create a simple calculator that does addition, subtraction, multiplication and division of real numbers using QLineEdit and QPushButton classes.

b. Further modify the program above to include input data validation.

Subject Code: MCA20208DCE
Subject Name: MANAGEMENT INFORMATION SYSTEM

Unit I

Organization and Information Systems, The Organization: Structure, Managers and activities – Data ,information and its attributes – The level of people and their information needs - Types of Decisions and information - Information System, - Management Information System (MIS) –Decision Support System (DSS) and Group Decision Support System (GDSS). (12 lectures)

Unit II

Need for System Analysis - Stages in System Analysis - Structured SAD and tools like DFD, Context level Diagram, Decision Table and Structured Diagram. System Development Models: Waterfall, Prototype, Spiral, –Roles and responsibilities of System Analyst, Database Administrator and Database Designer. (12 lectures)

Unit III

Enterprise Resources Planning (ERP): Features, selection criteria, merits, issues and challenges in Implementation - Supply Chain Management (SCM): Features, Modules in SCM – Customer Relationship Management (CRM): Phases. Knowledge Management and e-governance, Nature of IT decisions- Strategic decision. (12 lectures)

Unit 1V

Security and Ethical Challenges, Ethical responsibilities of Business Professionals – Business, technology, Computer crime – Hacking, cyber theft, unauthorized use at work. Issues and internet privacy. Challenges – working condition, health and social issues, Ergonomics and cyber terrorism.
(12 lectures)

RECOMMENDED/BOOKS:

1. “Management Information Systems”, Kenneth J Laudon, Jane P. Laudon, Pearson/PHI,10/e, 2007
2. “Management Information Systems”, W. S. Jawadekar, Tata McGraw Hill Edition, 3/e, 2004
3.Turban, Efraim, Ephraim McLean, and James Wetherbe. 2007. Information Technology for Management: Transforming Organizations in the Digital Economy. New York, John Wiley & Sons.

Subject Name: MANAGEMENT INFORMATION SYSTEM (Tutorials)

Week 1:

Q1. What is MIS and MIS definition.

Week 2:
Q2. Role and concept of MIS.

Week 3:

Q3. Objectives and Needs.

Week 4:

Q4. Types of Information System.

Week 5:

Q5. MIS Development Process: SDLC.

Week 6:

Q6. Ethical and Social issues.

Week 7:

Q7. Selection, Criteria and merits of ERP.

Week 8:

Q8. Challenges in implementation of ERP.

Week 9:

Q9. Features and modules of Supply Chain Management.

Week 10:

Q10. Phases of Customer Relationship Management.

Week 11:

Q11. Nature of IT decisions, Strategies.

Week 12:

Q12. Computer crime: Hacking, Cyber theft, unauthorized use at work.

Week 13:

Q13. Ergonomics and Cyber Terrorism.

General Elective Courses

Subject Code: MCA20209DCE

Subject Name: Python

Unit I

Understanding Python variables, Python basic Operators, python blocks , Data Types, Declaring and using Numeric data types: int, float, complex Using string data type and string operations[**4L**]

Defining list and list slicing Use of Tuple data type: Python Program Flow Control Conditional blocks using if, else and elif Simple for loops in python, For loop using ranges[**4L**]

String, list and dictionaries Use of while loops in python Loop manipulation using pass, continue, break and else Programming using Python conditional and loops block [**4L**]

Unit 2

Python Functions, Modules And Packages, Organizing python codes using functions[**4L**]

Organizing python projects into modules, Importing own module as well as external modules[**4L**]

Understanding Packages Powerful Lamda function in python, Programming using functions, modules and external packages, Python String, List And Dictionary Manipulations. [**4L**]

Textbook

- Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning, ISBN: 978-1111822705.

Reference Books:

1. David Beazley , Brian K. Jones “Python Cookbook”, 3rd Edition. O’Reilly Publications
2. Jake VanderPlas “Python Data Science Handbook” O’Reilly Publications
3. David Beazley, “Python Essential Reference (4th Edition) “ Addison Wesley
4. Vernon L. Ceder,” The Quick Python Book, Second Edition”, Manning Publications
5. Brett Slatkin ,”Effective Python”

Semester – III

Subject Code: MCA20301CR
Subject Name: Design and Analysis of Algorithms

Unit I

Introduction to Algorithms, Analysis of algorithms, Designing Algorithms, Growth of Functions, Asymptotic notations (4L)
Recurrences, Substitution method, Iteration method, Recursion trees (4L)
The Master Method, Time and Space Complexity study of some basic algorithms. (4L)

Unit II

Randomized Algorithms: Identifying the repeated element, Primality testing, Advantages and Disadvantages. (4L)
Divide and Conquer, General method, Binary search, Quick sort, Merge sort (4L)
Greedy Method, General method, Knapsack problem, Single source shortest paths.(4L)

Unit III

Dynamic programming, General methods, All pair shortest paths, Traveling salesman problems. (4L)
Backtracking, General method, 8-Queen problem, Sum of subsets, Knapsack problem.(4L)
Branch and Bound, General method, Least Cost Branch and Bound, 8-Queen Problem(4L)

Unit IV

Lower boundary theory, Lower bound theory through reductions, P and NP problems. NP hard and NP complete problems. (6L)

Approximate Algorithms and their need, The vertex Cover Problem,The traveling salesman problem, The subset sum problem (6L)

Text Book:

1. Horowitz, Sahni, Rajasekaran “Fundamentals of Computer Algorithms”, Galgotia Publications

Reference Books:

1. Coremen, Leiserson, Rivest,Stein, “Introduction to Algorithms”, 2nd edition, PHI.
2. Michael T. Goodrich, Roberto Tamassia “Algorithm Design and Applications” , Wiley
3. Aho, Hopcroft and Ullman, “The Design and Analysis of Computer Algorithms”, Pearson.

Subject Name: Design and Analysis of Algorithms (Tutorial)

Week 1. Illustrate with suitable examples the following divide and conquer algorithms

- Binary Search
- Quick Sort
- Merge Sort

Week 2. Analyze performance of above divide and conquer algorithm

Find an optimal solution to the Knapsack problem instance *n=3, m=20, (p1, p2, p3) = (25,24,15)* and *(w1, w2, w3) = (18,15,10)?*

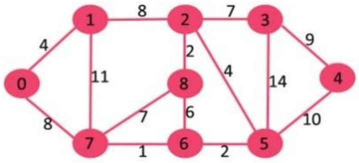
Week 3.Using dynamic programming method, find the optimal tour and its length (cost) for the directed graph whose edge lengths are represented by the following matrix?

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

Week 4. Trace the Quick sort algorithm on the following array of nine elements:

65, 70, 75, 80, 85, 60, 55, 50, 45

Week 5. Explain greedy (Dijkstra) algorithm for Single source shortest path using the following graph?



Week 6. Generate using backtracking algorithm the dynamic state space tree for the 0/1 Knapsack problem with following data: profit p = {11, 21, 31, 33, 43, 53, 55, 65}, weight w = {1, 11, 21, 23, 33, 43, 45, 55} and knapsack capacity m = 110 and n =8.

Recursions

Substitution method	Guess the form of the solution and prove it by induction
Iteration Method	Convert the recurrence into a summation and solve it
Master Method	Bound a recurrence of the form: $T(n) = aT(n/b) + f(n)$ $a \geq 1$, $b > 1$ 1. if $\mathbf{f(n) = O(n^{\log_b a-\varepsilon})}$, $\varepsilon > 0$ then $\mathbf{T(n) = \Theta(n^{\log_b a})}$

	2. if **$f(n) = \Theta(n^{\log_b a})$** then **$T(n) = \Theta(n^{\log_b a} \log n)$** 3. if **$f(n) = \Omega(n^{\log_b a+\varepsilon})$**, $\varepsilon > 0$ and $af(n/b) \le cf(n)$ for $c < 1$ then **$T(n) = \Theta(f(n))$**

Week 7.

$$T(1) = 1$$
$$T(n) = T\left(\tfrac{n}{2}\right) + \sqrt{n}$$

Prove that $T(n) = O(\sqrt{n})$ using the substitution method.

Week 8.

$T(a) = \Theta(1)$
$T(n) = T(n-a) + T(a) + n$
Find T(n) using the iteration method.

Week 9.

Use the Master method to find $T(n) = \Theta(?)$ in each of the following cases:

a. $T(n) = 7T(n/2) + n^2$
b. $T(n) = 4T(n/2) + n^2$
c. $T(n) = 2T(n/3) + n^3$

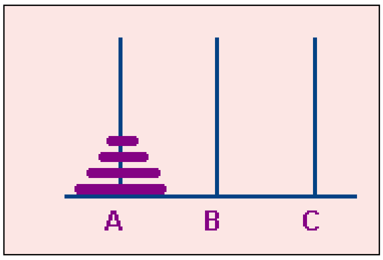
Week 10.

Fibonacci series is defined as follows:
$f(0) = 0$
$f(1) = 1$
$f(n) = f(n-1) + f(n-2)$

Find an iterative algorithm and a recursive one for computing element number n in Fibonacci series, Fibonacci(n).
Analyze the running-time of each algorithm.

Week 11.

***Hanoi towers problem*:**



n disks are stacked on pole A. We should move them to pole B using pole C, keeping the following constraints:

- We can move a single disk at a time.
- We can move only disks that are placed on the top of their pole.
- A disk may be placed only on top of a larger disk, or on an empty pole.

Analyze the given solution for the Hanoi towers problem; how many moves are needed to complete the task?

```
static void hanoi(int n, char a, char b, char c){
if(n == 1)
System.out.println("Move disk from " + a + " to "+ b +"\n");
else{
hanoi(n-1,a,c,b);
System.out.println("Move disk from " + a + " to "+ b +"\n");
hanoi(n-1,c,b,a);
   }
```

Week 12.

T(n) = 3T(n/2) + nlogn
Use the Master-Method to find T(n).

Week 13.

Matrix multiplication

Given two $n \times n$ matrix, A and B, give an efficient algorithm that computes its matrices multiplication

C=A·B, $C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$.

Analyze the time complexity of this algorithm.

Subject Name: Algorithms Lab

Week 1:

1. Write a program for Iterative Binary Search.
2. Write a program for Recursive Binary Search.

Week 2:

3. Write a program for Merge Sort.
4. Write a program for Quick Sort.

Week 3:

5. Write a program for single sources shortest path algorithm using greedy Method
6. Write a program for Floyde-Warshal algorithm.

Week 4:

7. Write a program for travelling salesman problem using Dynamic programming.
8. Write a program for linear Search

Week 5:

9. Write a program for Sort a given set of elements using heap
10. Implement 0/1 knapsack using dynamic programming

Week 6:

11. Print all the nodes reachable from a given starting node in a digraph using BFS method.
12. Implement Knapsack Problem using greedy Method

Week 7:

13. To implement 8-Queens problem and analyze its time complexity.
14. Implement BFS

Week 8:

15. Implement DFS
16. Write a program that finds shortest paths to other vertices using Dijkstra's algorithm, from a given vertex in a weighted connected graph

Week 9:

17. Write C programs to implement All-Pairs Shortest Paths problem using Floyd's algorithm.

16. Design and implement in C to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

Week 10:

17. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n > 5000 and record the time taken to sort. Plot a graph of the time taken versus n on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using C how the divide -and- conquer method works along with its time complexity analysis: worst case, average case and best case.

18.Implement recursive solution to the Tower of Hanoi puzzle.

Week 11:

18. Compute the transitive closure of a given directed graph using Warshall's algorithm

Subject Code: MCA20302CR
Subject Name: Java Programming

Unit I

Introduction to Java Language: Creation of Java. How Java changed the Internet. Features of Java Language. Evolution of Java. Comparison with other languages like C++.Java Virtual Machine (JVM) and Byte-code. [3L]

Java Language Overview: Lexical issues – Whitespace, Identifiers, Keywords, Literals, Separators, and Comments. Installing JDK.PATH variable. Java program – Structure, Compilation and Execution. Java Class libraries (System Class).*main()* method. [2L]

Data types, Variables and Arrays: Primitive Data-types and Typed-Literals. Variables – Declaration, Initialization, Scope and Lifetime. Arrays – Single and Multidimensional. Type Conversion and Expression Promotion. [4L]

Operators, Expressions and Control statements: Arithmetic, Bitwise, Relational, Logical, Assignment. Precedence and Associativity. Selection, Iteration and Jump Statements. [3L]

Unit II

Class Fundamentals: Class Structure (Variable and Method declaration).Modifiers (Access Modifiers and Other Modifiers).Components of Class, Variable and Method declaration. Constructor and *finalize()*. Garbage Collection. Passing parameters to methods. Variable hiding. Method overloading. Constructor overloading and chaining. Use of *this* keyword. Code blocks - Static and non-static. [4L]

Inheritance: Mechanism. Role of Access Modifiers. Method Overriding and Shadowing. Use of *super* keyword. Polymorphism - Early and Late binding. Abstract Class and Interface. Components of Interface declaration. Implementing Interfaces. [4L]

Exception Handling: Mechanism - Exception-Object, Throwing an Exception, and Exception Handler. Catch or Specify policy. Types of Exception - Checked vs Unchecked, Built-in vs User-defined. Catching an Exception - *try-catch-finally*. Specifying an Exception - *throws*. Manually throwing an Exception - *throw*. Custom Exceptions. Chained Exceptions. [4L]

Unit III

Packages: Creating and Importing Packages. CLASSPATH variable. *static import*. [2L]

Strings: Mutable and Immutable Strings. Creating Strings. Operations on Strings. [2L]

Threads: Creating Threads in Java. Java Thread Lifecycle. Multithreading in Java: Synchronization and Inter-process communication (IPC) in Threads. [4L]

Applet: Java Applet class Architecture. Working and Lifecycle of Java Applet. Displaying text and animation, and passing parameters to Applet. Embedding Applets in a web page. [4L]

Unit IV

Event-Driven Programming: Java 1.1 Event Delegation Model – Source object, Event object and Listener object. Methods associated with Source, Event and Listener objects. Low-level vs Semantic events. Adapter classes, Inner classes, and Anonymous Inner classes. Adding GUI elements to Applet. [5L]

I/O Streams: Byte, Character, Buffered, Data, and Object Streams. Standard Streams. File I/O Basics, Reading and Writing to Files. Serializing Objects. [4L]

Networking Classes and Interfaces: TCP/IP Server Sockets in Java. Developing simple networking applications in Java like File transfer, Chatting, etc. [3L]

Textbook: H. Schildt, *Java: The Complete Reference*, 9th Edition, Tata McGraw Hill, 2014.

Reference Books:

5. K. Sierra, *Sun Certified Programmer For Java 5*, Wiley India, 2006.
6. K. Sierra and B. Bates, *Head First Java (Java 5)*, 2nd Edition, O'Reilly, 2003.
7. H.M. Dietel and P.J. Dietel, *Java: How to Program*, 6th Edition, Pearson Education, 2007.
8. C.S. Horstmann and G. Cornell, *Java 2 Vol-1 Fundamentals*, 7th Indian Reprint, Pearson Education, 2006.
9. E. Balagurusamy, *Programming with Java: A Primer,* 4th Edition, Tata Mcgraw Hill, 2010.

Subject Name: Java Programming (Tutorials)

Week 1.

a. Compare features and advancements offered by various versions of Java SE (starting from Java 1 to Java 8).

b. Prepare a timeline that highlights introduction of most important features of Java.

c. Prepare a mind-map that records all the steps of Java Development Kit installation.

Week 2.

a. Write a Java program that reads an integer from keyboard and displays it.

b. Write a Java program that reads a floating-point number from keyboard, converts it to integer and displays it.

c. Write a Java program that reads a string from keyboard, converts it to a floating-point number and displays it.

d. Write a Java program that populates all the 10 elements of an integer array using keyboard input, increments every element by 1, and displays every element.

e. Write a Java program that iteratively calculates factorial of a number.

Week 3.

a. Write a Java program that creates a Class with public instance variables and accesses them (sets and retrieves value) using dot operator.

b. Write a Java program that creates a Class with private instance variables and accesses them (sets and retrieves value) using public methods.

c. Write a Java program that creates a Class with public static variables and accesses them (sets and retrieves value) using Class name.

d. Write a Java program that creates a Class with private static variables and accesses them (sets and retrieves value) using public static methods.

Week 4.

a. Write a Java program that creates an instance of a Class that has an explicit constructor, and *static* code blocks and *non-static* code blocks each displaying a different string.

b. Write a Java program that creates a Class that has an explicit constructor, and *static and non-static* code blocks each displaying a different string. Extend this class into another class that also has an explicit constructor, *static code blocks and non-static* code blocks, each displaying a different string. Create an instance of the derived class.

c. Write a Java program that creates a Class with *2 overloaded constructors* each displaying a different string. Extend this class into 2 classes each invoking a different constructor of their super-class when instantiated.

d. Write a Java program that has a private instance variable (say *int x*) and a public method with an argument of same type and name as the instance variable (i.e. *int x* in this case). Access (set and retrieve values) both variables inside the method.

Week 5.

a. Write a Java program that creates a Class in which a method throws *ArithmeticException,* and handles it by displaying a relevant message.

b. Write a Java program that creates a Class in which a method throws *ArrayOutOfBoundsException*, and handles it by displaying a relevant message.

c. Write a Java program that creates a Class in which a method throws *FileNotFoundException*, and handles it by displaying a relevant message.

Week 6.

a. Write a Java program that uses fully qualified name for *System.out.println()* method to display a string.

b. Create a Java Package (say *pack1*) that contains 2 Classes (say A and B) wherein each class has a *show()* method that displays a unique string. Use all the classes from the package in another Java program and invoke the *show()* method.

c. Write a Java program that demonstrates use of *static import.*

Week 7.

a. Write a Java program that reads a string from keyboard and displays the number of characters read without using any built-in method.

b. Write a Java program that reads a string from keyboard and displays the number of alphabets, numerals and special characters in it.

c. Write a Java program that reads a string and a character from keyboard, and splits the read string into multiple strings based on the character read.

d. Write a Java program that reads a string from keyboard and displays it in reverse.

e. Write a Java program that reads a string from keyboard and displays its first character.

Week 8.

a. Write a Java program to create a Class that extends a *Thread* class. Use this class to start 2 threads that displays unique strings.

b. Write a Java program to create a Class that implements *Runnable* interface. Use this class to start 2 threads that displays unique strings.

c. Write a Java program that creates many threads and obtains the reference of all these threads to display their name and priority.

d. Write a Java program that demonstrates use of *synchronized* keyword.

e. Write a Java program that demonstrates use of *wait()* and *notify()* methods.

Week 9.

a. Write a Java program to create a Class that extends an Applet class and displays a message that illustrates the order in which various methods of Applet class are executed.

b. Write a Java program to create a Class that extends an Applet class. The derived Applet accepts a string from the keyboard and displays it.

c. Write a Java program to create a Class that extends an Applet class. The derived Applet is passed a string as a parameter that it displays.

d. Write a Java program to create a Class that extends an Applet class. The derived Applet displays numbers from 1 to 10, one after another with a time gap of 1000ms between evry two displayed numbers.

e. Write a Java program to create a Class that extends an Applet class. The derived Applet takes length and breadth as parameters and draws a rectangle.

Week 10.

a. Write a Java program to create a Class that extends an Applet class. The derived Applet should also have a *TextField* widget and a *Button* widget. The *TextField* should display the current date when the *Button* is pressed.

b. Write a Java program to create a Class that extends an Applet class. The derived Applet accepts two numbers from keyboard via *TextField* widgets and displays their addition or subtraction in a third *TextField* widget depending upon whether "Add" or "Sub" *Button* widget is pressed.

Week 11.

a. Write a Java program that accepts a filename (passed as command line argument) and checks whether it exists or not, and if it exits then it verifies whether it is a directory or a regular file.

b. Write a Java program that opens a file (filename is passed as command line argument), and displays its attributes.

c. Write a Java program that opens and displays the contents of a file (filename is passed as command line argument).

d. Write a Java program that opens a file (filename is passed as command line argument) and appends "hello world!" at its end.

Week 12.

a. Write a Java program (a client) that opens a connection to https://www.Internic.net website and displays information about www.google.com.

b. Write a Java program (Client) that sends a text message to another Java program (Server), and the Server displays an acknowledgement message on receiving it.

c. Write a Java program (Client) that sends a text string to another Java program (Server), which receives it and sends back the reverse string of the received string.

Subject Name: Java Programming Lab

Week 1. This week students will learn how to acquire and install Java Development Kit.

a. Download latest version of Java Development Kit (JDK), preferably JDK8 or above (Please visit https://java.com/en/download/).

b. Follow the instructions that appear during the Installation ofJDK8, and set PATH variable to the appropriate directory location as instructed in the lecture.

Week 2. This week students will learn how to write, compile and execute simple Java programs.

a. Write a Java program that displays "*hello world!"* on *the* screen.

b. Write a Java program that receives two integer numbers via keyboard, does their summation, and displays the result. Ensure that only integer values are processed.

c. Write a Java program that prints the season name corresponding to its month number using *If-else* and *switch-case* statements.

d. Write a Java program that sorts (using *bubble sort*) an integer array using *for* loop.

e. Write a Java program that calculates factorial of a number (inputted via keyboard) recursively.

f. Write a Java program that creates a 2D integer array with 5 rows and varying number of columns in each row. Using '*for each'* variant of *for* loop display each element of every row.

Week 3. This week students will learn how to define and instantiate a Class, and access its members.

a. Write a Java program that creates a Class, namely *Student*.

   i. Ensure that *Age* instance variable of the Class is never accessed directly, and its value is never less than 4 and greater than 40 for any Object of the Class (use methods to validate and assign the value).

   ii. Ensure that the constructor always assigns a unique value to *Enrollment_No* instance variable for every Object of the Class (use a *static* class variable for counting objects, say *Object_Counter*).

   iii. Ensure that when an Object is removed, the *Object_Counter* is automatically decremented (use *finalize()*), and whenever required the variable can only be accessed using a method even without an Object reference (make the counter private and use a *static* method to access it).

b. Write a Java program in which a Class overloads a method *sum()*, which takes 2 parameters. The overloaded methods should perform summation of either integer or floating-point values.

Week 4. This week students will learn the principles of Inheritance and polymorphism in Java.

a. Write a Java program that creates a Class namely*A* that has a private instance variable and method, a protected instance variable and method, a default instance variable and method, and a public instance variable and method. Create another Class say *B* that inherits from *A*.

i. Show that all except private members are inherited.

ii. Show that an inherited instance variable can be shadowed (with the same or weaker access visibility) but can be accessed using *super* keyword in the sub-class.

iii. Show that an inherited method can be overridden (with the same or weaker access visibility) but canbe accessed using *super* keyword in the sub-class.

iv. Show that the reference variable of type A or B can't access an overridden method of A in the Object of B.

v. Show that the reference variable of type A can access a shadowed data member of A in the Object of B.

Week 5. This week students will learn the principles of Exception Handling in Java.

a. Write a Java program that creates a Class in which a method asks the user to input 2 integer values, and calls another member function (say *div()*) to divide the first inputted number by the second number (by passing them as parameters). Handle an exception that can be raised in *div()* when the denominator equals zero (use *try-catch* statement).

b. Modify the above Java program so that it also creates a Custom Exception that is thrown by *div()* when the denominator value is 1 (use *throw*). Handle the exception.

c. Modify the above Java program so that the exception-handling in not performed by *div()* rather it only species all the possible exceptions it may throw (use *throws*). And, the method that calls *div()* does the exception handling.

Week 6. This week students will learn how to create and import Packages in Java.

a. Create a Java Package (say *pack1*) that contains 3 Classes (say A, B and C). Write a Java program that uses this package after setting the CLASSPATH variable. Following scenarios must be considered individually:

i. Importing the whole package (all the 3 classes)

ii. Importing only specific class (say Class A only)

b. Create another Package (say *pack2*) that contains same number of classes, and same definition for each class, as that of *pack1*. Write a Java program that imports all classes from both *pack1* and *pack2* while ensuring that the name conflicts are not encountered while accessing any of these classes.

Week 7. This week students will learn about String operations in Java.

a. Write a Java program to count the number of words in a string that is passed as a command line argument.

b. Write a Java program to check whether a string is palindrome or not.

c. Write a Java program to count the total number of occurrences of a given character in a string.

d. Write a Java program to convert a string to char array.

Week 8. This week students will learn how to create and manage Threads in Java.

a. Write a Java program that creates a Class that extends a *Thread* class. Create 3 objects of the class, each starting a new thread and each thread displaying "*I am Thread:* " in an infinite loop. The displayed text must be suffixed by the unique name of the thread.

b. Write a Java program that creates a Class that implements interface *Runnable*, and does the same as the above program.

c. Write a Java program to implement a solution for producer-consumer problem using synchronization and inter-process communication in Threads.

Week 9. **This week students will learn how to create and use Applets in Java.**

a. Write a Java program that creates a Class that extends an Applet class. The applet is embedded in a web page and is passed 2 numeric parameters. The applet shall display the summation result of the parameters passed.

b. Write a Java program that creates a Class that extends an Applet class. The applet simulates a marquee by displaying characters of the message one at a time from right to left across the screen. When the message is fully displayed, the message starts again.

c. Write a Java program that creates a Class that extends an Applet class. The applet displays bar chart forthe data passed as parameter. The data includes the number of male and female students enrolled in MCA course.

Week 10. **This week students will learn the principles of Event-Driven Programming in Java.**

a. Write a Java program that creates a Class that extends an Applet class. Add GUI elements to the applet so as to create a simple 2-player tic-tac-toe game.

b. Write a Java program that creates a Class that extends an Applet class. Add GUI elements to the applet so as to create a simple calculator.

Week 11. **This week students will learn how to use I/O Steams in Java.**

a. Write a Java program to open and read a file (filename is passed as command line argument), and displays the number of words in the file?

b. Write a Java program to copy a file. The source and destination filenames arepassed as command line arguments.

Week 12. **This week students will learn how to use networking classes in Java.**

a. Write a Java program (client) that sends a text message to another Java program (server), which receives and displays it.

b. Modify the above Java programs so that each of the two programs is able to send and receive the text messages.

Subject Code: MCA20303DCE

Subject Name: OPERATING SYSTEMS

Unit I: Operating System Concepts Review

Operating System concepts, Types; Operating System Structures – Processes, Scheduling criteria, Scheduling Algorithms, Interprocess Communication and Synchronization, Classical problems, Critical section, Semaphores, Monitors **5 Lectures**

Deadlocks characterization, Methods for handling deadlocks, Deadlock - Prevention, Avoidance, Detection, Recovery **3 Lectures**

Memory Management: Address Spaces, Virtual Memory, Page Replacement Algorithms, Design and Implementation Issues for Paging Systems, Segmentation **4 Lectures**

Unit II: Distributed Operating Systems Concepts-I

Client Server model; Remote procedure call and implementation issues; Processor allocation and scheduling in distributed systems - System Models, Load balancing and sharing approach, fault tolerance; Real time distributed systems **4 Lectures**

Synchronization in Distributed Systems - Clock Synchronization and related algorithms, Logical Clocks. **4 Lectures**

Mutual Exclusion: Centralized & Distributed (Contention & Token) Algorithms. Election Algorithms: Bully Algorithm, Invitation Algorithm & Ring **4 Lectures**

Unit III: Distributed Operating Systems Concepts-II

Deadlock Detection - Distributed Algorithms; Threads - Characteristics, Advantages & Disadvantages, Design Issues & Usage. **4 Lectures**

General architecture of Distributed Shared Memory systems; Design and implementation issues of DSM; granularity - Structure of shared memory space, consistency models, replacement strategy, thrashing. **4 Lectures**

Filesystems: Hierarchical filesystems, Distributed Filesystem (DFS): goals, Requirements, Components, Service Types. Hadoop-DFS: Design & Issues. **4 Lectures**

References:

Abraham Silberchatz, Peter B. Galvin, Greg Gagne, "Operating System Principles", John Wiley.
Andrew .S. Tanenbaum, "Modern Operating Systems", PHI.
Andrew. S. Tanenbaum, "Distributed Operating System", PHI.

Subject Name: OPERATING SYSTEMS
Practice Sheets for Tutorials/Practical's

Week 1:	**Write a program to demonstrate the use of process system calls provided by LINUX:** • **fork()** • **getpid()** **The program should display the process ids for both the parent and the child.**
Week 2:	**Write a program to simulate the SJF scheduling algorithm.** **The program should read the following inputs:** • **Number of processes** • **Burst time of each process** **The program should generate the following outputs:** • **Process statistics after each context switch** • **Average Turnaround time** • **Average Waiting time**
Week 3:	**Write a program to simulate the Round Robin scheduling algorithm.** **The program should read the following inputs:** • **Number of processes** • **Burst time of each process** • **Length of the Time Slice** **The program should generate the following outputs:** • **Process statistics after each context switch** • **Average Turnaround time** • **Average Waiting time**
Week 4:	**Write a program to simulate the FIFO page replacement algorithm.** **The program should read the following inputs:** • **Length of the reference string** • **Reference string** • **Number of page frames** **The program should generate the following outputs:** • **Page replacement sequence after each reference** • **Number of page faults**
Week 5:	**Write a program to simulate the LFU page replacement algorithm.** **The program should read the following inputs:** • **Length of the reference string** • **Reference string** • **Number of page frames** **The program should generate the following outputs:** • **Page replacement sequence after each reference** • **Number of page faults**
Week 6:	**Write a program to simulate the LRU page replacement algorithm.** **The program should read the following inputs:** • **Length of the reference string** • **Reference string** • **Number of page frames** **The program should generate the following outputs:** • **Page replacement sequence after each reference** • **Number of page faults**
Week 7:	**Create client server programs using RPC wherein the server accepts a number from the client and returns the square of the number which is then displayed by the client.** **Use *rpcgen* to generate the stubs automatically.**
Week 8:	**Create a server program which returns the current date time to a client through RPC. Use the same basis as used in week 7. You will need the following functions on the server:**

	• time() • ctime()
Week 9:	**Write a program to simulate Clock Synchronization in Distributed Systems using Lamport's Algorithm.**
Week 10:	**Write a program to implement mutual exclusion of threads on LINUX using the pthread.h library Some of the important system calls to be used include:** • **pthread_mutex_lock, pthread_self, pthread_create, pthread_exit**
Week 11:	**Write a program to simulate the BULLY ELECTION ALGORITHM.**
Week 12:	**Write a program to simulate the Banker's Algorithm for Deadlock Avoidance.** **The program should read the following inputs:** • **Number of Processes** • **Number of resource types** • **Current allocation and Maximum allocation of resources to each process** • **Currently Available Resources** • **New request details** **The program should generate the following outputs:** • **Determine whether the system is in the safe state or not** • **If it is, then determine the safe sequence**
Week 13:	**Write a set of programs to use the concept of shared memory through LINUX system calls.** • **One process creates a shared memory segment and writes a message into it.** • **Another process opens the segment, reads the message and outputs the message to standard output.** **Some of the important system calls to be used include:** • **shmget(), shmat(), shmctl() etc.**

Subject Code: MCA20304DCE
Subject Name: Organisational Behaviour

Organizational Behaviour

Introduction to Organizations and Individuals. What is an organization, components of organization, nature and variety of organizations (in terms of objectives, structure etc.), models of analysing organizational phenomena, organizational and business variables, organizations in the Indian context, institutions and structures, basic roles in an organization, etc., perception, attitudes, motives (achievement, power and affiliation), commitment, values creativity and other personality factors, profile of a manager and an entrepreneur.

Interpersonal and Group Processes - Interpersonal trust, understanding the other person from his/her point of view, interpersonal communication, listening, feedback, counselling, transactional analysis, self-fulfilling prophecy, etc., leadership, motivating people, working as a member of a team, team functioning, team decision-making, team conflict resolution, team problem solving.

Organizational Structure and Integrating Interpersonal and Group Dynamics-Elements of structure, functions of structure, determinants of structures, dys functionalities of structures, structure-technology environment-people relationships, principles underlying design of organizations; organizational culture, organizational politics, issues of power and authority, organizational communications, organizational change, integrating cases(s).

Case method and lectures should be supplemented with a variety of other methodologies such as feedback on questionnaires and tests, role plays, and behaviour simulation exercise.

References

Arnold, John, Robertson, Ivan T. and Cooper, Cary, L., "Work Psychology: Understanding Human Behaviour in the Workplace", MacMillan India Ltd., Delhi, 1996.
Dwivedi, R.S., "Human Relations and Organisational Behaviour: A Global Perspective", MacMuillan India Ltd., Delhi, 1995. "
Arnold, John, Robertson, Ivan T. and Cooper, Cary, L., "Work Psychology: Understanding Human Behaviour In the Workplace", MacMillan India Ltd., Delhi, 1996.
Dwivedi, R.S., "Human Relations and Organisational Behaviour: A Global Perspective", MacMillan India Ltd., Delhi, 1995.
French and Bell (4th ed), "Organization Development: Behavioral Science Interventions for Organization Improvement", Prentice Hall of India Pvt. Ltd., New Delhi, 1994.
Hellriegel, Slocum and Woodman, "Organizational Behaviour", West Publishing Co. USA, 1986.
Hersey and Blanchard (6th 00), "Management of Organizational Behaviour: Utilising Human Resources", Prentice Hall of India Pvt. Ltd., New Delhi, 1996.
Prasad, Kesho, "Organisational Development for Excellence", MacMillan India Ltd., New Delhi, 1996.
Robbins (4th 00), "Essentials of Organizational Behaviour", Prentice Hall of India Pvt. Ltd. New Delhi, 1995.
Schermerhorn, Hunt and Osborw, "Managing Organization Behaviour", John Willey & Sons, USA. 1982.
Weston, Mergers, "Restructuring and Corporate Control", Prentice Hall of India Pvt. Ltd. New Delhi, 1995.

Discipline Centric Elective Courses

Subject Code: MCA20305DCE
Subject Name: Theory of Computation

UNIT I

Introduction to computation, Finite Automata, DFA, Kleene's theorem, Non determinism, Finite Automata with output.

Regular Languages: introduction to formal languages, regular operations, closure property

Regular Expression; Equivalence of DFA, NFA, and RE. Non-Regular Languages and Pumping Lemma.

Unit II

Context-Free Languages: introduction to CFL, context free grammars, Chomsky normal form, parse trees, derivation and ambiguity, closure and non-closure properties.

Pushdown Automata (PDA), Deterministic vs Non-deterministic PDAs. Non-CFL and pumping Lemma for CFLs

UNIT III

Context-Sensitive Languages: introduction to CSL, context sensitive grammars, Linear Bounded Automata (LBA)

Recursive and Recursively Enumerable Languages: introduction to REL and Chomsky hierarchy, Hilbert's algorithm and Church-Turing Thesis.

Turing Machines, equivalence of Deterministic, Non-deterministic, and multi-tape TMs. Universal TMs.

Unit IV

Decidable Languages: Decidability, Relationship between Solvability and Decidability, halting problem;

UnDecidability, Complexity: Asymptotic Notation and properties thereof. Reductions and its applications. Deterministic and Non-deterministic Turing Machine cost models (space and time)

References:

Cohen, Daniel IA, and Daniel IA Cohen. *Introduction to computer theory*. Vol. 2. New York: Wiley.

Linz, Peter. *An introduction to formal languages and automata*. Jones & Bartlett Learning.

Parkes, Alan P. *Introduction to languages, machines and logic: computable languages, abstract machines and formal logic*. Springer Science & Business Media, 2012.

Subject Name: Theory of Computation (Tutorials)

Week 1:

1. Prove that $(w^R)^n = w$ for all w belonging to $\sum^*$?
2. Let L= {ab,aa,baa}.Which of the following strings are in L*= { abaabaaabaa, aaaaabaaaa, baaaaaabaaaab, baaaaabaa}.?
3. Show that (L*)* = L* for all languages?
4. Give a simple description of the language generated by the grammar with productions?
   i. S →aA,
   ii. A→bS,
   iii. S→ λ
5. What language does the grammar with these productions generate
   i. S→ Aa
   ii. A→B
   iii. B→Aa
6. Prove that

$(L1L2)^R = L2^R L1^R$

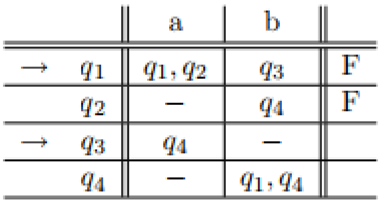
For all languages L1 & L2.

Week 2:

1. Design a simple sequence detector for the sequence 011. Include three outputs that indicate how many bits have been received in the correct sequence.
2. Design a sequence detector that searches for a series of binary inputs to satisfy the pattern 01[0*]1, where [0*] is any number of consecutive zeroes?
3. What is the difference between Moore & Mealy machine
4. How are transducers different from other automata?

Week 3:

1: Convert the nondeterministic automaton given below to an equivalent deterministic one using the subset construction. Omit inaccessible states. Draw the graph of the resulting DFA.



	a	b	
→ q_1	q_1, q_2	q_3	F
q_2	–	q_4	F
→ q_3	q_4	–	
q_4	–	q_1, q_4	

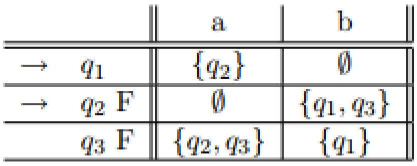
2: Give a deterministic finite automaton over the alphabet {a, b} which accepts all strings containing no more than two consecutive occurrences of the same input letter. (For example, abba should be accepted but not abaaab.)

3: Consider the following unary operation on languages:

min(L) = {x ∈ L | no proper prefix of x is in L}

Prove that regular languages are closed under this operation; that is, prove that if language A is regular, then so is min(A).

4: . Convert the nondeterministic automaton given below to an equivalent deterministic one using the subset construction. Omit inaccessible states. Draw the graph of the resulting DFA



	a	b
→ q_1	$\{q_2\}$	$\emptyset$
→ q_2 F	$\emptyset$	$\{q_1, q_3\}$
q_3 F	$\{q_2, q_3\}$	$\{q_1\}$

Week 4:

1. Construct npda's that accept the following regular languages
    a. L1 = L(aaa*b)
    b. L1 = L(aab*aba*)
2. Construct an NPDA that accepts language L = $\{a^n b^m : n>=0, n \neq m\}$.
3. Consider the language:

$L = \{a^m b^n \mid m \neq n\}$

Give a simple argument for L being context–free?

4. Construct an NPDA (possibly with €–transitions) accepting L on empty stack. Explain its workings.

Week 5:

1: Find a Context-Free Grammar for the following language

$$L = \{a^n b^m c^k : k = n + m \}$$

2: Find a CFG that generates the language

$$L(G) = \{ a^n b^m \mid 0 \leq n \leq m \leq 2n\}.$$

3: Which language generates the grammar G given by the productions

S → aSa | aBa
B → bB | b

4: Consider the grammar

S → abScB | λ
B → bB | b

What language does it generate?

Week 6:

1: Apply the Pumping Lemma for context–free languages (as a game with the Demon) to show that the language:

$$A = \left\{a^n b^n a^j \mid j \leq n\right\}$$

Is not context free.

2: Which language generates the grammar G given by the productions S → aSa | aBa, B → bB | b

3: Consider the grammar
S → abScB | λ
B → bB | b
What language does it generate?

4: Find a CFG that generates the language

$$L(G) = \{ a^n b^m \mid 0 \leq n \leq m \leq 2n\}.$$

Week 7:

1: Give a detailed description of a total Turing machine accepting the palindromes over {a, b}: that is, all strings x ∈ {a, b} ∗ such that x = rev x.
2: Suppose we try to construct a Turing machine to solve a particular problem, but we are not successful. Does this mean that no Turing machine exists that can solve that problem? Explain and justify your answer.
3: Draw the state diagram for a Turing machine that increments a binary number. Assume that the input tape contains at least one non-blank symbol.
4: Construct a Turing machine that accepts

$$L = \left\{ww : w \in \{a, b\}^+\right\}.$$

Week 8:

1: Let L and L’ be decidable languages. Prove the following properties.

i. The complement L’ is decidable

2: Prove that L ∪ L’ is decidable when L is decidable

3: Consider the problem of determining whether a two-tape Turing machine ever writes a non-blank symbol on its second tape, i.e.

$$N = \{\langle M, w\rangle \mid M \text{ is a two-tape Turing machine which writes a non-blank symbol onto its second tape when it runs on } w\}.$$

Show that N is undecidable.

4: Is there an explicit program P so that for a given y it is decidable whether P terminates on input y?

Course No.: MCA20306DCE
Course Title: Wireless and Mobile Communication

Unit I

Classification and types of Wireless telephones. Introduction to Cordless, Fixed Wireless (WLL), Wireless with limited mobility(WLL-M) and (Fully)Mobile Wireless phones. Introduction to various generations of mobile phone technologies and future trends. Wireline vs. Wireless portion of mobile communication networks. Mobile-Originated vs. Mobile-Terminated calls. Mobile Phone numbers vs. Fixed-Phone numbers.

Unit II

Concept of cells, sectorization, coverage area, frequency reuse, cellular networks & handoffs. Wireless Transmission concepts; types of antennas; concepts of signal propagation, blocking, reflection, scattering & multipath propagation. Comparison of multiple access techniques FDM, TDM and CDM. Concept of Spread Spectrum(SS) techniques; Frequency Hopping SS . Direct Sequence SS and concept of chip-sequence.

Unit III

Concept of Forward and Reverse CDMA channel for a cell/sector. Concept/derivation of Walsh codes & Code Channels within a CDMA Channel. Simplified illustration of IS-95 CDMA using chip sequences. Purpose of Pilot, Sync, Paging, Forward Traffic Channels. Purpose of Access & Reverse TCs.

Unit IV

GSM reference architecture and components of Mobile Networks: MS, BTS, BSC, MSC; their basic functions and characteristics. Use of HLR and VLR in mobile networks. Handoff scenarios in GSM.

References Books:

T. Rappaport, “Wireless Communications, Principles and Practice(2nd Edition)”,Pearson.Andy Dornan, “The Essential Guide to Wireless Communications Applications”,Pearson. Jochen Schiller, “Mobile Communications”, Pearson. K.Pahlavan, P.Krishnamurthy, “Principles of Wireless Networks”, PHI.

Subject Code: MCA20307DCE
Subject Name: Cloud Computing

UNIT I

CLOUD COMPUTING FUNDAMENTALS [10L]

Cloud Computing: Introduction, Key characteristics, goals of cloud computing; Benefits of Cloud Computing: flexibility, scalability, portability; Drawbacks of cloud computing: dependability, security, reliability, availability

Major components of cloud computing: application, client, infrastructure, platform, service, storage, processing power

Private, public and hybrid cloud. Cloud computing platforms: IaaS, Example Amazon EC2; PaaS, Example Google App Engine, Microsoft Azure; SaaS;

UNIT II

CLOUD ECONOMICS [10L]

Cloud Economics: Economics of private clouds, Software productivity in the cloud, Economies of scale: public vs. private clouds: Economics of choosing a Cloud platform for an organization, based on application requirements, economic constraints and business needs, SLA document, Virtualization and Resource Provisioning in Clouds

UNIT III

CLOUD TECHNOLOGIES & CLOUD SECURITY [10L]

Technologies and the processes required when deploying web services: Web services: XML, SOAP, REST, SOAP versus REST, AJAX: asynchronous ‘rich’ interfaces, Mashups: user interface services

Security issues, Solutions to threats, Disaster recovery

UNIT IV

CLOUD DEVELOPMENT [10L]

Data in the cloud: Relational databases; Cloud file systems: GFS and HDFS; BigTable, HBase;

Cloud data stores: Datastore and SimpleDB

Dev 2.0 platforms: Salesforce.com’s Force.com platform, TCS InstantApps on Amazon cloud; Advantages, applicability and limits of Dev 2.0

Emerging Cloud Platforms

REFERENCES

1. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge University Press; 1 edition, [ISBN: 9780521137355], 2010.
2. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach” McGraw-Hill Osborne Media; 1 edition [ISBN: 0071626948], 2009.
3. Dimitris N. Chorafas, “Cloud Computing Strategies” CRC Press; 1 edition [ISBN: 1439834539],2010.

Subject Code: MCA20308DCE
Subject Name: Machine Learning

Unit I

Canonical ML problems (Regression, Binary Classification, Multiclass Classification, Clustering, Ranking); Supervised learning, unsupervised learning, reinforcement learning etc. [3L]

Decision Trees: introduction, pruning, rule extraction from trees, learning rules from data. [2L]

Classification Algorithms: KNN and effect of various distance algorithms (Euclidean, Manhattan, Mahalanobis Distances, etc.). [3L]

Clustering Algorithms: K-mean clustering, fuzzy K-means, hierarchical clustering. [3L]

Dimensionality: curse of dimensionality, feature pruning, and normalization. [2L]

Unit II

Principal Component Analysis: by variance maximization, by Mean Square Error Compression, for dimensionality reduction [4L]

Discriminant analysis: Linear Discriminant Analysis and Quadratic Discriminant Analysis [4L]

Support Vector Machines: margins, optimal margin classifiers and Lagrange duality, kernels, regularization, Sequential minimal optimization algorithm. [5L]

Unit III

Ensemble Classifiers: Boosting (AdaBoost), random ensembles (random forests). [4L]

Model Based Prediction (Bayes): estimating distribution parameters from data, Use of Bayes theorem to identify optimal classifier. [5L]

Mixture Models: GMM, Maximum likelihood estimation of parameters, Expectation Maximization algorithm, EM for GMM. [4L]

Reference Books:

1. Introduction to Machine Learning by EthemAlpaydin, MIT Press

2. Pattern Classification by Duda and Hart. John Wiley publication

3. The Elements of Statistical Learning: Data Mining, Inference, and Prediction by Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer.

4. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer

5. Machine Learning: A probabilistic Perspective, by Kevin P. Murphy, MIT Press

Laboratory Experiments to be carried out by students.

Required: Python and/or MATLAB, scikit Learn

LAB1&2: Implement a Decision Tree (use IRIS dataset) for classification tasks. Use cross-entropy as a quality-criterion

a. What is an appropriate stopping criterion?
b. Give the probability (histogram) at each decision step (for each class).
c. Plot the cross entropy of each attribute (at each decision step).
d. What is the Information gain at each new node created?
e. Display the complete tree (use breadth first approach).
f. Use post-pruning approach to restrict the decision tree complexity (split dataset: 34% for train, 33% for validation, and 33% testing). That is, show, for each pruning decision, the gain at the parent node and collective gain of the children. How does the gain of post-pruning compare with that of part d?

LAB3&4: Implement and visualize KNN, using a dataset (IRIS or some other dataset). Describe the effects of various parameters of KNN algorithms on the performance.

a. How do different data splitting's effect the performance (say: 70% Train, 30% Test vs others)?
b. How do various distance measures perform (Manhattan, Euclidean etc.)
c. Write a function to predict the class for a given input value.
d. How does the value of K affect the performance and how doe we choose and optimal value of K?

LAB5: Implement and visualize K-means, using a dataset (faithful dataset or some other dataset). Describe the effects of various parameters of K-means algorithms on the performance.

LAB6: Build a support vector machine (SVM) for classification, say spam. Use a dataset (one which is libsvm format) with scikit for classification. How does svm compare with a previous modal (use on Decision tree or KNN or KMEANS)?

LAB7&8: Use scikit Learn to compare LDA and PCA for Iris dataset.

a. Plot the sample of first two principal components (that is the direction for most variance in the dataset).
b. For LDA, which are the attributes that account for most variance in the dataset?
c. How does LDA and PCA help in dimensionality reduction(for the dataset in use)?
d. Implement Fischer's LDA from scratch.
e. Implement a classification function for the LDA (that for a given input sample for the dataset it should identify its class).

LAB9: Use some dataset(handwritten digits) and show how a Random Forest can be used as a classifier.

LAB10&11: Naïve Bayes (implement from scratch for some dataset)

a. How does the algorithm work?
b. What are the variations to Naïve Bayes?
c. What are the pros and cons of using the algorithm?
d. What are the steps required to implement a basic Naïve Bayes?
e. How can we improve the Naïve Bayes?
f. When should we use Naïve Bayes?

Lab 12&13: Implement a GMM clustering

a. What is the weakness in K-means that motivates us to use GMM?
b. For some given dataset (where clusters are near to each other) visualize how K-means vs GMM, perform clustering.
c. Implement an Expectation Maximizing algorithm.
d. Implement a prediction function, which returns the class/cluster to which an input sample belongs.
e. Show how GMM can be used to estimate the probability distribution of a dataset.

General Elective Courses

Subject Code: MCA20309DCE
Subject Name: Software Project Management

Managerial Issues in Software Projects:

Introduction to software markets; Planning of software projects; Size and Cost Estimations; Project Scheduling; Measurement of software quality and productivity; ISO and Capability Maturity Models for organisational growth.

Project management and Practice.

Managing the systems life cycle; requirements determination, logical design, physical design. testing, implementation; system and database integration issues; metrics for project management and systems performance evaluation, managing expectations; superiors. users. team members. and other related to the project; determining skill requirements and staffing the project; cost-effectiveness analysis; reporting and presentation techniques; and effective management of both behavioural and technical aspects of the project.

References

- Gilb, T., "Principles of Software Engineering Management", Addison Weskey. Reading. M.A .• 1988.
- Putnam. L.H . Myers. W., "Industrial Sire"" Software - Effective Management using Measurement". IEEE C.S. Press. 1997.