



2019-20

*Look forward,
learn modern
knowledge, and
do not waste
time in studies
of old subjects of
no values.*



Lab Manual M.C.A. SEMESTER - II



CSM-22P1: LABORATORY COURSE - II

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY ALIGARH

CREDITS

Lab Manual Design Committee:

- Prof. Mohammad Ubaidullah Bokhari*
- Dr. Arman Rasool Faridi*
- Dr. Faisal Anwar*
- Dr. Aasim Zafar (Convener)*

Editor:

- Dr. Aasim Zafar*

Design & Compilation:

- Mr. Sunil Kumar Sharma*

First Edition: *July 2017*

Approved by BoS, Dated-31-07-2017

Department of Computer Science, A.M.U., Aligarh, (U.P.), India

COURSE TITLE: Laboratory Course-II
CREDIT: 04
CONTINUOUS ASSESSMENT: 40

COURSE CODE: CSM–22P1
PERIODS PER WEEK: 06
EXAMS: 60

COURSE DESCRIPTION

This course is designed to help students in learning JAVA using object-oriented paradigm. Approach in this course is to take JAVA as a language that is used as a primary tool in many different areas of programming work, programming of Numerical and Statistical methods/techniques using MATLAB and introducing database management language.

COURSE CONTENT

This course is designed to provide the students the opportunity of learning the differences between C++ and JAVA programming, to develop, debug, and execute JAVA programs.

Further, the concept of MATLAB is introduced with emphasis to implement various numerical and statistical methods.

The concept of database management system is introduced and students learn to design and create database, emphasize on data dictionaries, normalization, data integrity, data modelling, and creation of simple tables, queries etc.

OBJECTIVES

This course is designed to help students in:

- ❑ Gaining knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.

- ❑ Understanding the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc and exception handling mechanisms.
- ❑ Learning the use of arrays, access protection, wrapper classes etc.
- ❑ Understanding the principles of inheritance, packages and interfaces.
- ❑ Learning to create applet and application and event handling, AWT controls, able to create GUI (frame, menu, button, text boxes, layout manager).
- ❑ Familiarizing with the important topics and principles of software development.
- ❑ Learning the ability to write a computer program to solve specified problems.
- ❑ Understanding to use the Java SDK environment to create, debug and run simple Java programs.
- ❑ Getting the fundamental knowledge and practical abilities in MATLAB required to effectively utilize this tool in technical numerical computations and visualization in other courses.
- ❑ Introducing the students to the software MATLAB for numerical computations and in particular familiarizing them with the MATLAB Desktop, basic commands through the Command window.
- ❑ Understanding the different issues involved in the design and implementation of a database system.
- ❑ Studying the physical and logical database designs, database modeling, relational, hierarchical, and network models
- ❑ Understanding and using data manipulation language to query, update, and manage a database
- ❑ Developing an understanding of essential DBMS concepts such as: database security, integrity, and concurrency.

- ❑ Designing and building a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

OUTCOMES

After completing this course, the students would be able to:

- ❑ Understand the fundamental features of an object oriented language like JAVA: object classes and interfaces, exceptions and libraries of object collections.
- ❑ Understand how to implement, compile, test and run JAVA programs, comprising more than one class, to address a particular software problem.
- ❑ Understand how to include arithmetic operators and constants in a JAVA program and able to deploy JAVA applet and application program.
- ❑ Understand the concept of multithreading, multitasking, and multiprogramming.
- ❑ Understand the use of members of classes found in the JAVA API (such as the Math class, Wrapper classes).
- ❑ Understand the implement the keyboard/mouse events.
- ❑ Understand the concept of object oriented programming, inheritance, constructor, interfaces, and package.
- ❑ Understand the concept of Exception Handling, Files and Streams, Applets and Graphics and also learn the concept of Applet classes, Applet life cycle and JAVA Swing (introductory part).
- ❑ How to take the statement of a business problem and from this determine suitable logic for solving the problem; then be able to proceed to code that logic as a program written in JAVA.
- ❑ Use MATLAB for interactive computations.

- Familiar with memory and file management in MATLAB.
- Program scripts and functions using the MATLAB development environment.
- Use basic flow controls (if-else, for, while) in MATLAB.
- Enhance the knowledge and understanding of Database analysis and design.
- Enhance the knowledge of the processes of Database Development and Administration using SQL and PL/SQL.
- Enhance Programming and Software Engineering skills and techniques using SQL and PL/SQL.
- Design and implement normalized database structures.
- Define program-data independence, data models for database systems, database schema and database instances.
- Identify Structure Query Language statements used in creation and manipulation of Database.
- Identify the methodology of conceptual modeling through Entity Relationship model.
- Identify the methodology of logical & physical model.
- Preparation of background materials and documentation needed for Technical Support using SQL and PL/SQL.
- Use the Relational model and how it is supported by SQL and PL/SQL.
- Develop an understanding of the differences between OODBMS, ORDBMS and RDBMS and the practical implications of each approach.
- Analyze and design a real database application.
- Solve Database problems using Oracle, SQL and PL/SQL. This will include the use of Procedures, Functions, Packages, and Triggers.

RULES AND REGULATIONS

Students are required to strictly adhere to the following rules.

- ❑ The students must complete the weekly activities/assignments well in time (i.e., within the same week).
- ❑ The students must maintain the Lab File of their completed activities/assignments in the prescribed format (**Appendix-1**).
- ❑ The students must get the completed weekly activities/assignments checked and signed by the concerned teachers in the Lab in the immediate succeeding week. Failing which the activities/assignments for that week will be treated as incomplete.
- ❑ At least **TEN (10)** such timely completed and duly signed weekly activities/assignments are compulsory, failing which students will not be allowed to appear in the final Lab Examination.
- ❑ The students need to submit the following three deliverables for each exercise duly signed by the Teacher:
 - ❖ Coding
 - ❖ Input /Output
- ❑ The students need to ensure that each question is assessed and signed by the Teacher in the week/time.
- ❑ Late submission would not be accepted after the due date.
- ❑ Cooperate, collaborate and explore for the best individual learning outcomes but copying is strictly prohibited.

APPENDIX-1

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#			
	2#			
	3#			
2	1#			
	2#			
	3#			
3	1#			
	2#			
	3#			
4	1#			
	2#			
	3#			

Note: The students should use Header and Footer mentioning their roll no. & name in footer and page no in header.

WEEK #1

OBJECTIVES

- To help the students in understanding the importance of programming in Object Oriented environment using JAVA
- To help the students in understanding the basic structure of JAVA Program
- To help students getting familiar with JAVA programming environment and IDE.
- To learn the students in writing, debugging and executing JAVA programs.
- How to read input from keyboard?
- To learn the constant and variables.
- To learn the concepts of operators used in JAVA.

OUTCOMES

After completing this, the students would be able to:

- write, debug and execute simple JAVA programs.
- use the constant and variables and operators in JAVA.

PROBLEMS

- 1# Write a java program to print '*Hello World*'.
- 2# Write a java program to print the corresponding address of a student.
- 3# Write a Java program to calculate sum of two numbers.
- 4# Write a Java program to convert the given temperature in Fahrenheit to Celsius using the following conversion formula $C = (F - 32) / 1.8$ and display the value in a tabular form.

5# Write a java program for finding the sum, difference, product, quotient, minimum and maximum of any two integers.

6# Write a java program 'MyNumber.java' that performs following operations on a variable 'num' of type double:

i) Finds the round value of 'num' and stores the result in a variable numRound of type double.

ii) Finds the ceil value of 'num' and stores the result in a variable numCeil of type double.

iii) Finds the floor value of 'num' and stores the result in a variable numFloor of type double.

Cast 'num' to type int and stores the result in a variable numInteger of type int. Display output of numRound, numCeil, numFloor and numInteger on screen.

Note: Test your program with following value of num

num=56.764 num=36.432

7# Write Java program to show uses of all Math class methods.

WEEK #2

OBJECTIVE

- To learn using JAVA programming coding: data types, variable, constants, operators, Control Statement (*if, switch, loops*)
- To learn break, continue statements, ternary operator, bit-wise operators, user-defined data types in JAVA, order of evaluation of different operators in java.
- To learn the controls statements and loops.

OUTCOMES

After completing this,

- The students would be able to use different control statements and loops available in JAVA.

PROBLEMS

- 1# Write a java program to prints the count of odd and even no's entered.
- 2# Write a java program to print the squares and cubes for the numbers 1 to 5.
- 3# Write a java program that computes the sum of the reciprocals:

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots \dots \dots + \frac{1}{10}$$

- 4# Write Java program to compute the sum of the 2+4+6+-----N Terms.

- 5# Shown below is a Floyd's triangle:

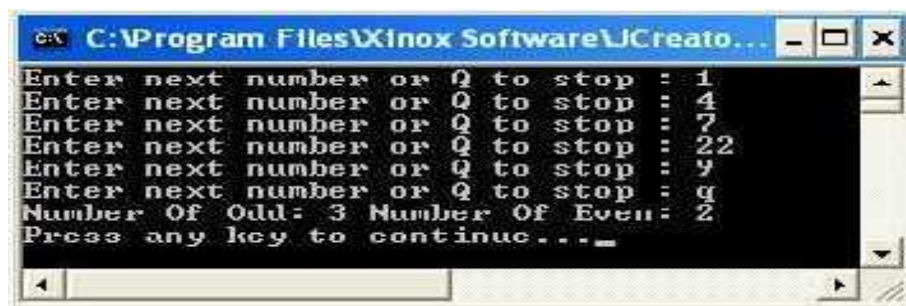
```
1
2 3
4 5 6
7 8 9 10
```

11 12 13 14
15 16 17 18 19

- i) Write a program to print the above triangle.
- ii) Modify the program to produce the following form of Floyd's triangle.

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

- 6# Write a program in the sequence 0, 1, 1, 2, 3, 5, 8, 13, 21 is called *Fibonacci numbers*. Write a program using a *do...while* loop to calculate and print the first *m Fibonacci numbers*.
- 7# Write a program to accept three digits (i.e. 0 - 9) and print all its possible combinations. (For example if the three digits are 1, 2, 3 than all possible combinations are: 123, 132,213, 231, 312, 321)
- 8# Write a program to calculate the area of triangle, square, circle, rectangle by using method overloading.
- 9# Write a Java Program which prompts the user to enter 4 numbers. The program will then computes and display their sum and their product.
- 10# Write a Java program which reads a 4-digit number and prints the digits on separate lines. (Each digit is printed on one line).Output of your program must be in the following format:



11# The intersection method computes the intersection of two rectangles- that is, the rectangle that is formed by two overlapping rectangles: You call this method as follows:

Rectangle r3 =r1.intersection (r2);

Write a program that constructs two rectangle objects, prints them, and then prints their intersection. What happens when the rectangles do not overlap?

WEEK #3

OBJECTIVES

- ❑ To learn the object oriented features (classification, encapsulation, inheritance, polymorphism, abstraction) and java features like secure, platform independent, portable, threading.
- ❑ To learn the concept of class like how to declare a class, how to define methods inside the class, how the access the properties of base class into derived class, Code reusability.
- ❑ To learn the use of arrays, access protection, wrapper classes.
- ❑ To learn the use of this keyword, use of toString () method.

OUTCOME

After completing this, the students would be able to:

- ❑ employ various types of selection constructs in a JAVA program
- ❑ demonstrate the hierarchy of JAVA classes to provide a solution to a given set of requirements.

PROBLEMS

1# Write Java program involving two classes: *OddAndEven* & *TestOddAndEven*.

OddAndEven has the following:

-) Instance variables countOfOdd and countOfEven both of type int.
-) A method addNumber that takes a number as parameter and increment countOfOdd, if the number is odd, else increment countOfEven.
-) A method toString that returns a string in the form: “Number of Odd: x, Number of Even: y”, where x and y are the values of the instance variables.

The *TestOddAndEven* class first creates *OddAndEven* object, then in a loop, read a number and use it to call the *addNumber* method until the user enters Q. Finally, it prints the count of odd and even numbers entered.

2# Design a class *Circle* and implement the following methods:

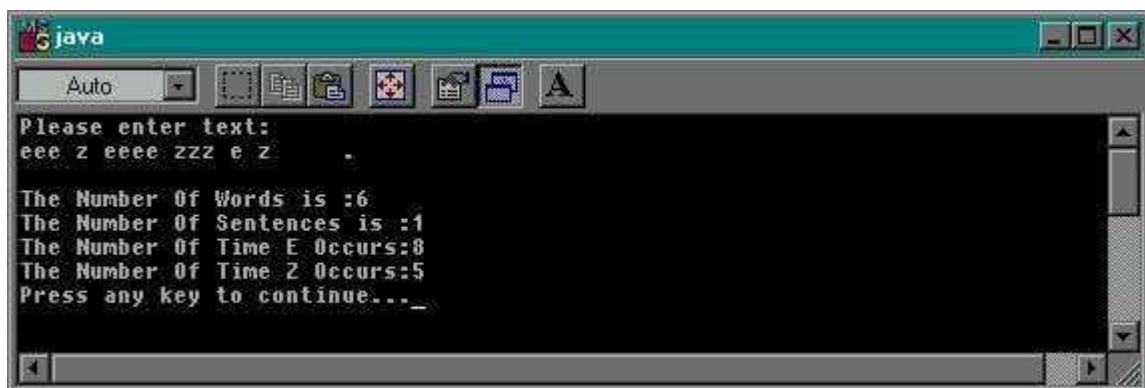
-) Define a circle method to compute its area
-) Define a circle method to compute its perimeter
-) Define a method that takes a given point represented by a pair of numbers and checks whether or not the point is inside the circle.

The circle class needs to have instance variables to store the radius of the circle, and the x and y coordinates of the center. Add main program to test the class *Circle* repeatedly, until user enters negative value for the radius of the circle.

3# Write a program in Java that reads in text and prints as output the following:

-) The number of words in the text
-) The number of sentences in the text
-) The number of times the letter “e” occurs in the text
-) The number of times the letter “z” occurs in the text

(Note: Use *StringTokenizer* class)



```
java
Auto
Please enter text:
eee z eeee zzz e z
The Number Of Words is :6
The Number Of Sentences is :1
The Number Of Time E Occurs:8
The Number Of Time Z Occurs:5
Press any key to continue...
```

4# A sales person is paid commission based on the sales he makes as shown by the following table:

Sales	Commission
Under ₹500	2% of Sales
Between ₹500 to ₹5000	5% of Sales
₹5000 and Above	8% of Sales

Write a class *Commission* which has an instance variable *sale*, an appropriate constructor, and a method *commission ()* that returns the commission.

Now write a demo class to test the *Commission* class by reading a sale from the user, using it to create a *Commission* object after validating that the value is not negative. Finally, call the *commission ()* method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

5# The certain instructor assigns letter grade for his course based on the following table:

Score	Grade
>=90	A+
>=85	A
>=80	B+
>=75	B
>=65	C+
>=60	C
>=55	D+
>=50	D
<50	F

Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate method *letterGrade()* that returns the letter grade as a String.

Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater than 100. Finally, call the *letterGrade()* method to get and print the grade.

6# Implement a Student class with the following fields, constructors and methods:

Fields:

```
) name;  
) totalScore;  
) numberOfQuizzes;
```

Constructors:

```
) public Student(String name, double score)  
) public Student(double score, String name)  
) public Student(String name)
```

Methods:

```
) public String getName()  
) public double getAverage() //this should return zero if no quiz has been  
    taken.  
) public double getTotalScore()  
) public void addQuiz(double score)  
) public void printStudent() //this should print the student's name and  
    average score.  
) public String toString()
```

Write an application TestStudent that reads a student name and use the Student class to create a Student object. Then read the scores of the student in three quizzes and add each to the totalScore of the student using addQuiz() method and print the student object.

(Note: Make use of this key word wherever it can be used).



```
java
Auto
Enter name of Student: Usamah
Enter quiz #1 for Usamah: 20
Enter quiz #2 for Usamah: 17
Enter quiz #3 for Usamah: 19
Name: Usamah, Average: 18.666666666666668
Press any key to continue...
```

7# Write a program to design a class to represent a bank account. Include the following members.

Date members:

-) Name of depositor
-) Account Number
-) Type of account
-) Balance account in the account

Methods:

-) To assign initial values
-) To deposit an account
-) To withdraw an account after checking balance.
-) To display the name and balance

WEEK #4

OBJECTIVES

- To learn the concept Interface, how to implement it,
- To learn the use of member functions and how we access them, using interfaces for code reusing,
- To learn converting between class and interface types, using interfaces for callbacks; Polymorphism, Inheritance
- To learn the Divide and Conquer Method.

OUTCOMES

After completing this, the students would be able to:

- handle interfaces, converting between class and interface types, callbacks, Polymorphism, Inheritance, Inheriting instance fields and methods.
- access the member functions of a class.
- understand the concept of Divide and Conquer method and implement it in different application areas.

PROBLEMS

- 1# Write a program that reads in a sentence from the user and prints it out with each word reversed, but with the words and punctuation in the original order:



```
C:\Program Files\Xinox Software\JCreator Pro\GE2001.exe
Input : I said to him: Go ye, now!
Output: I diae ot mih: oG ey, won!
Press any key to continue...
```

- 2# Write a program where interface can be used to support multiple inheritances. Develop a standalone Java program for this.

- 3#** Write a program that reads in three strings and sorts them lexicographically.
Hint: Enter strings: Charlie Able Banker
Output: Able Banker Charlie
- 4#** Implement the classes for the shapes using an interface for the common methods, rather than inheritance from the super class, while still Shape as a base class.
- 5#** Implement a super class Person. Make two classes, Student and Instructor, inherit from Person. A person has a name and a year of birth. A student has a major, and an instructor has a salary. Write the class definitions, the constructors, and the methods toString for all classes. Supply a test program that tests these classes and methods.
- 6#** Define a class Employee having private members – id, name, department, salary. Define default and parameterized constructors. Create a subclass called “Manager” with private member bonus. Define methods accept and display in both the classes. Create n objects of the Manager class and display the details of the manager having the maximum total salary (salary+bonus).
- 7#** Write a Java program to create a super class Vehicle having members Company and price. Derive 2 different classes *LightMotorVehicle* (members – mileage) and *HeavyMotorVehicle* (members – capacity-in-tons). Accept the information for n vehicles and display the information in appropriate form. While taking data, ask the user about the type of vehicle first.
- 8#** Implement Binary Search in C/C++ using Divide and Conquer approach.
- 9#** Implement Merge Sort in C/C++ using Divide and Conquer approach.
- 10#** Implement Quick Sort in C/C++ using Divide and Conquer approach.

WEEK #5

OBJECTIVES

- To learn the concept of Access control: private access, public access, protected access and package access.
- To help students in getting familiar with MATLAB programming Environment and IDE.
- To learn writing, debugging and executing simple MATLAB programming.

OUTCOMES

After completing this, the students would be able to:

- understand the concept of Access control, its purpose, its characteristics, how many types of Access control.
- write, debug and execute a simple MATLAB program.

PROBLEMS

- 1# Write a MATLAB program to print natural number from 1 to 100.
- 2# A bank maintains two kinds of accounts - Savings Account and Current Account. The savings account provides compound interest, deposit and withdrawal facilities. The current account only provides deposit and withdrawal facilities. Current account holders should also maintain a minimum balance. If balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number, and type of account. From this derive the classes Curr-acct and

Sav-acct. Include the necessary methods in order to achieve the following tasks:

-) Accept deposit from a customer and update the balance.
-) Display the balance.
-) Compute interest and add to balance.
-) Permit withdrawal and update the balance (Check for the minimum balance, impose penalty if necessary).

3# Define a class called fruit with the following attributes:

-) Name of the fruit
-) Single fruit or bunch fruit

Define a suitable constructor and *displayFruit()* method that displays values of all the attributes. Write a program that creates 2 objects of fruit class and display their attributes.

4# Write a program where interface can be used to support multiple inheritances. Develop a standalone Java program for this.

5# Implement the classes for the shapes using an interface for the common methods, rather than inheritance from the super class, while still Shape as a base class.

6# Create a package called "Arithmetic" that contains methods to deal all arithmetic operations. Also, write a program to use the package.

7# Write a program to make use of a parameterized method inside a class. Take the following case: Create a class Box and define a method in this class which will return the volume of the box. Initialize two objects for your class and print out the volumes respectively.

WEEK #6

OBJECTIVES

- To learn the concept of Exception handling.
- To learn the importance of exceptions.
- To learn the concept and way of throwing exceptions.
- To decipher use of MATLAB in programming.
- To learn the practical use of MATLAB tool.
- To learn how to write and run a basic MATLAB program.
- To learn the concepts of Taylor series used in MATLAB.
- To learn the concept of Greedy Method.

OUTCOMES

After completing this, the students would be able to:

- understand the features of Exception handling.
- understand the concept of Exception handling.
- use throwing exceptions.
- use MATLAB for solving numerical problems.
- understand the concept of Greedy method and implement it in different applications areas.

PROBLEMS

- 1# Write a program that calls a method that throws an exception of type *ArithmeticException* at a random iteration in a *for* loop. Catch the exception in the method and pass the iteration count when the exception occurred to the calling method by using an object of an exception class you define.
- 2# Write a MATLAB program to estimate the sine function as a Taylor series.
- 3# Write a program that will count the number of character in a file

- 4#** In a small firm employee numbers are given in serial numerical order, that is 1, 2, 3 etc. Write a menu driven program to perform following operations:
- i) Create a file of employee data with following information: Employee No, Name, Sex, Gross Salary.
 - ii) Append the data of a new employee joining the firm.
 - iii) If a given employee leaves, delete his record.
 - iv) If gross salary of a given employee increases, update the gross salary.
 - v) Display the record of :
 - a. a given employee or
 - b. all employees.
- 5#** Write a program to create a sequential file that could store details about five products. Details include product code, cost, and number of items available and are provided through the keyboard.
- 6#** Implement Minimum Cost Spanning Tree in C/C++ by Prim's Algorithm.
- 7#** Implement Minimum Cost Spanning Tree in C/C++ by Kruskal's Algorithm.

WEEK #7

OBJECTIVES

- To learn the file handling using Java.
- To learn various operations performed in file using file handling concepts.
- To learn implementation of SIMPSON'S 1/3 rule method using MATLAB.
- To identify the methodology of conceptual modeling through Entity Relationship model.
- To understand and identify the methodology of logical & physical model.

OUTCOMES

After completing this, the students would be able to:

- work with Files and Streams.
- use MATLAB for solving definite integrals using SIMPSON'S 1/3 rule method.
- Identify the methodology of conceptual modeling through Entity Relationship model.
- Understand and identify the methodology of logical & physical model.

PROBLEMS

- 1# Write a Java program which reads student grades from a text file called grades.txt and prints only the corresponding letter grades into a file called letter.txt. The letter grades are assigned according to the following table. Assume that the grades.txt file can have any number of students' grades. Hint: The last number in the grades.txt file is -1

Score	Grade
≥ 90	A+
≥ 85	A
≥ 80	B+
≥ 75	B
≥ 65	C+
≥ 60	C
≥ 55	D+
≥ 50	D
< 50	F

- 2#** Write a MATLAB program to find integral values $y(x) = 1/(1+x)$ within limits 0 to 6 using SIMPSON'S 1/3 RULE.
- 3#** Write a program to read a, b, c from data file and store roots of the quadratic equation in output file. You must open your output file in append mode.
- 4#** Develop an applet that receives three numeric values as input from the user and then displays the largest of the three on the screen. Write a HTML pages and test the applet.
- 5#** Consider a CONFERENCE_REVIEW database in which researchers submit their research papers for consideration. Reviews by reviewers are recorded for use in the paper selection process. The database system caters primarily to reviewers who record answers to evaluation questions for each paper they review and make recommendations regarding whether to accept or reject the paper. The data requirements are summarized as follows:

- a) Authors of papers are uniquely identified by e-mail id. First and last names are also recorded.
- b) Each paper is assigned a unique identifier by the system and is described by a title, abstract, and the name of the electronic file containing the paper.
- c) A paper may have multiple authors, but one of the authors is designated as the contact author.
- d) Reviewers of papers are uniquely identified by e-mail address. Each reviewer's first name, last name, phone number, affiliation, and topics of interest are also recorded.
- e) Each paper is assigned between two and four reviewers. A reviewer rates each paper assigned to him or her on a scale of 1 to 10 in four categories: technical merit, readability, originality, and relevance to the conference. Finally, each reviewer provides an overall recommendation regarding each paper.
- f) Each review contains two types of written comments: one to be seen by the review committee only and the other as feedback to the author(s).

Design ER diagram and construct database for the above. Give logical reasoning for the database design.

WEEK #8

OBJECTIVES

- To learn the concept of applet and their applications.
- To learn the concept of Trapezoidal method.
- To learn the concept of SIMPSON'S 3/8 rule.
- To learn the concept of Dynamic programming.
- To manipulate database using database language.
- To enhance the knowledge and understanding of Database analysis and design.
- To identify Structure Query Language statements used in creation and manipulation of Database.

OUTCOMES

After completing this, the students would be able to:

- understand applets in terms of drawing graphical shapes, colours, fonts, drawing complex shapes, reading text input inside applet.
- understand the concepts of Trapezoidal method.
- understand the concepts of SIMPSON'S 3/8 rule.
- understand the concepts of Dynamic programming and implement it through various real life problems.
- understand the Database analysis and design.
- use SQL in creating and manipulating the Database.

PROBLEMS

1# Write applets to draw the following shapes:

a) Cone b) Cylinder c) Square inside a circle

2# Write a MATLAB program to find integral values $y(x) = x^3$ within limit 0 to 1 considering five sub intervals using Trapezoidal method.

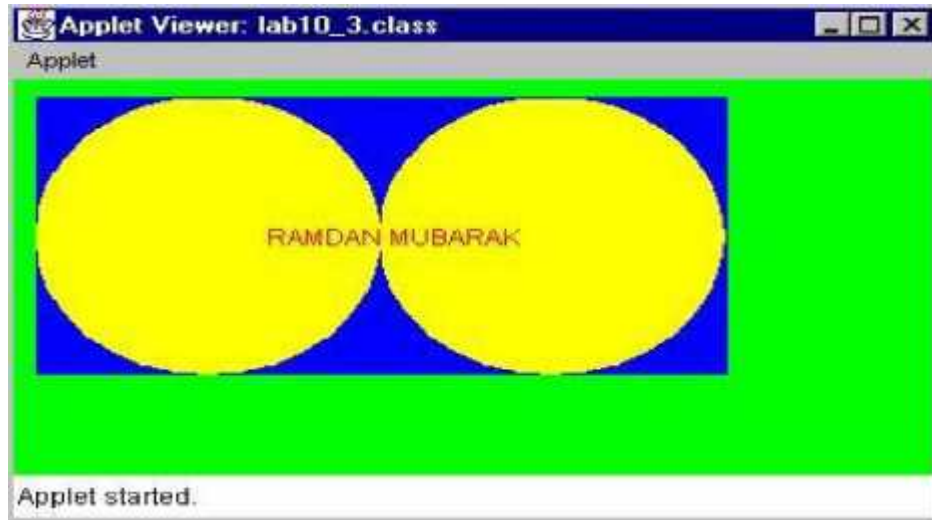
3# Write an applet that will display the following on a green background. Use the following dimension:

Rectangle : (10, 10, 300, 150), Fill colour: blue

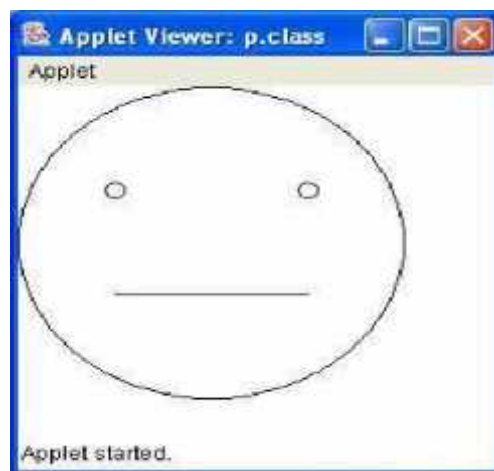
Left Circle : (10, 10, 50, 150), Fill Colour: Yellow

Right Circle: (159, 10, 150, 150), Fill colour: Yellow

Text : (110, 90), colour: Red



4# Write a JAVA Applet program to plot the following face:



5# Write a MATLAB program to find integral values $y(x) = \int_0^1 dx/(1+x)$ taking $h=1/6$ using SIMPSON'S 3/8 rule.

- 6# Implement Single Source shortest Path for a graph (Dijkstra , Bellman Ford Algorithm) in C/C++.
- 7# Implement all pair of Shortest path for a graph (Floyd-Warshall Algorithm) in C/C++.
- 8# Implement Travelling Salesman Problem in C/C++.
- 9# Create the tables described below:

Table Name: **CLIENT_MASTER**

Description: Used to store client information.

Column Name	Data Type	Size	Default	Attributes
CLIENTNO	Varchar2	6		
NAME	Varchar2	20		
ADDRESS 1	Varchar2	30		
ADDRESS 2	Varchar2	30		
CITY	Varchar2	15		
PINCODE	Number	8		
STATE	Varchar2	15		
BALDUE	Number	10,2		

Table Name: **PRODUCT_MASTER**

Description: Used to store product information.

Column Name	Data Type	Size	Default	Attributes
PRODUCTNO	Varchar2	6		
DESCRIPTION	Varchar2	15		
PROFITPERCENT	Number	4,2		
UNITMEASURE	Varchar2	10		
QTYONHAND	Number	8		
REORDERLVL	Number	8		
SELLPRICE	Number	8,2		

COSTPRICE	Number	8,2		
-----------	--------	-----	--	--

Table Name: **SALESMAN_MASTER**

Description: Used to store salesman information working for the company.

Column Name	Data Type	Size	Default	Attributes
SALESMANNO	Varchar2	6		
SALESMANNAME	Varchar2	20		
ADDRESS 1	Varchar2	30		
ADDRESS 2	Varchar2	30		
CITY	Varchar2	20		
PINCODE	Number	8		
STATE	Varchar2	20		
SALAMT	Number	8,2		
TGTTGET	Number	6,2		
YTDSALES	Number	6,2		
REMARKS	Varchar2	60		

Insert the following data into their respective tables:

Data for CLIENT_MASTER table:

ClientNo	Name	City	Pincode	State	BalDue
C00001	Ivan Bayross	Mumbai	400054	Maharashtra	15000
C00002	Mamta Muzumder	Madras	780001	Tamil Nadu	0
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
C00005	Hansel Colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	560050	Karnataka	0

Data for PRODUCT_MASTER table:

ProductNo	Description	Profit Percent	Unit Measure	QtyOn Hand	ReorderLvl	SellPrice	CostPrice
P00001	T-Shirts	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	500	350

P06734	Cotton Jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	350	250
P07975	Lycra Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

Data for SALESMAN_MASTER table:

SalesmanNo	Name	Address1	Address2	City	PinCode	State
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra

SalesmanNo	SalAmt	TgtToGet	YtdSales	Remarks
S00001	3000	100	50	Good
S00002	3000	200	100	Good
S00003	3000	200	100	Good
S00004	3500	200	150	Good

a) Exercise on retrieving records from a table

- i. Find out the names of all the clients.
- ii. Retrieve the entire contents of the Client_Master table.
- iii. Retrieve the list of names, city and the state of all the clients.
- iv. List the various products available from the Product_Master table.
- v. List all the clients who are located in Mumbai.
- vi. Find the names of Salesman who have a salary equal to Rs.3000.

b) Exercise on updating records in a table

- i. Change the city of ClientNo 'C00005' to 'Bangalore'.
- ii. Change the BalDue of ClientNo 'C00001' to Rs. 1000.
- iii. Change the cost price of 'Trousers' to Rs. 950.00.
- iv. Change the city of the salesman to Pune.

c) Exercise on deleting records in a table

- i. Delete all salesman from the Salesman_Master whose salaries are equal to Rs. 3500.
 - ii. Delete all products from Product_Master where the quantity on hand is equal to 100.
 - iii. Delete from Client_Master where the column state holds the value 'Tamil Nadu'.
- d) Exercise on altering the table structure
- i. Add a column called 'Telephone' of data type 'number' and size='10' to the Client_Master table.
 - ii. Change the size of SellPrice column in Product_Master to 10,2.
- e) Exercise on deleting the table structure along with the data
- i. Destroy the table Client_Master alongwith its data.
 - ii. Exercise on renaming the table
 - iii. Change the name of the Salesman_Master table to sman_mast.

WEEK #9

OBJECTIVES

- To learn the concept of applet and their applications.
- To learn the basic concepts of Simultaneous Linear Equations
- To learn the practical use of Iterative Methods viz. Bisection.
- To learn the concept of Backtracking.

OUTCOMES

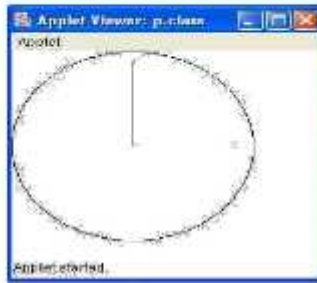
After completing this, the students would be able to:

- understand applets in terms of drawing graphical shapes, colours, fonts, drawing complex shapes, reading text input inside applet.
- access to the Iterative methods.
- understand the concept of Backtracking and implement it in various real life problems.

PROBLEMS

- 1# Write a graphics program that draws a clock face with a time that the user enters in a text field. (The user must enter the time in the format hh:mm, for example 09:45).

Hint: You need to find out the angles of the hour hand and the minute hand. The angle of the hour hand is harder; it travels 360 degree in 12 x 60 minutes. Your output must be in the following format:



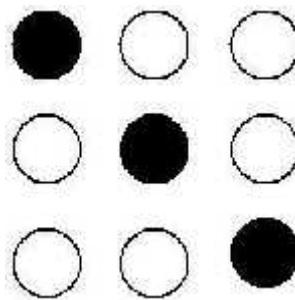
2# Write a program that draws the picture of a house.

Sample Output:



3# Write an applet to display the following figure:

Sample Output:



4# Write a MATLAB program to find the real root of the equation $x \log_{10} x = 1.2$ using Bisection method correct to four decimal places.

5# Implement Graph Colouring Problem in C/C++.

6# Implement Hamiltonian Problem in C/C++.

WEEK #10

OBJECTIVES

- To learn the concept of applet and their applications.
- To learn the practical use of Iterative Methods viz. False Position.
- To enhance the knowledge of the processes of Database Development and Administration using SQL and PL/SQL.

OUTCOMES

After completing this, the students would be able to:

- understand applets in terms of drawing graphical shapes, colours, fonts, drawing complex shapes, reading text input inside applet.
- access to the Iterative method.
- understand processes of Database Development and Administration using SQL and PL/SQL.

PROBLEMS

- 1# Draw a “bull’s eye” a set of concentric rings in alternation black and white colours: Fill a black circle, and then fill a smaller white circle on top, and so on.



- 2# Write a program to find a real root of the equation $3x + \sin x - e^x = 0$ using the method of false position correct to four decimal places.
- 3# Write a MATLAB program to find the root of the equation $xe^x = \cos x$ in the interval (0, 1) using Regula-Falsi method correct to four decimal places.
- 4# Assume that you are developing a prototype database of the AMU library management system, for that you need to create the following tables:

Book Records

Book details

Member details and

Book issue details

Structure of the tables is given below:

<i>Table Name</i>	<i>Attribute Name</i>
<i>Book Records</i>	<i>Accession Number</i>
	<i>ISBN Number</i>
<i>Books</i>	<i>ISBN Number</i>
	<i>Author</i>
	<i>Publisher</i>
	<i>Price</i>
<i>Members</i>	<i>Member Id</i>
	<i>Member Name</i>

Maximum Number of books that can be issued

Maximum Number of days for which book can be issued

Book Issue Member Id

Accession Number

Issue Date

Return Date

You must create constraints, including referential integrity constraints, as appropriate. Please note accession number is unique for each book. A book, which has no return date, can be considered as issued book. Enter suitable data into the tables. Now answer the following:

- a) Insert data in all the three tables (use insert).
- b) Insert appropriate description associated with each table and the column (use comment).
- c) Display the structure of the tables.
- d) Display the comments that you have inserted.
- e) Using SELECT statement, write the queries for performing the following function:
 - i. Get the list of all books (No need to find number of copies)
 - ii. Get the list of all members
 - iii. Get the Accession number of the books which are available in the library
 - iv. On return of a book by a member calculate the fine on that book.
 - v. List of books issued on 01-Jan-2005
 - vi. Get the list of all books having price greater than Rs. 500/-

- vii. Get the list of members who did not have any book issued at any time.
 - viii. Get the list of members who have not returned the book.
- f) Display member ID and the list of books that have been issued to him/her from time to time.
 - g) Find the number of copies of each book (A book accession number would be different but ISBN number would be the same).
 - h) Find the number of copies available of a book of given ISBN number.
 - i) Get the member ID and name of the members to whom no more books can be issued, because they have already got as many books issued as the number for which they are entitled.

WEEK #11

OBJECTIVES

- To learn the basic concepts of Transcendental Equations and how it can be solved.
- To learn the practical use of Iterative Methods viz. Newton-Raphson iterations etc.
- To learn and implement the different methods of least square and curve fitting using MATLAB.
- To learn and implement of different interpolation methods for finding the approximation using MATLAB.
- To learn the concepts of Graph Traversal.

OUTCOMES

After completing this, the students would be able to:

- Solve and find the real root of transcendental equations using iterative methods.
- Implement the different methods of least square and curve fitting using MATLAB.
- Find the approximation of data for given set of values using MATLAB.
- Understand the concepts of Graph Traversal and implement various algorithms of it.

PROBLEMS

- 1# Write a MATLAB program to find the real root of the equation $3x = \cos x + 1$ correct to four decimal places Using Newton-Raphson method.
- 2# Write a MATLAB program to implement Curve Fitting to fit a straight line.

3# Write a MATLAB program to implement Curve Fitting to fit a parabola.

4# Given below is the list of marks obtained by a class of 50 students in annual examination.

43,65,51,27,79,11,56,61,82,09,25,36,07,49,55,63,74,81,49,37,40,49,16,75,87,
91,33,24,58,78,65,56,76,67,45,,36,63,12,73,49,51,19,39,49,68,93,85,59,54

Write a program to count the number of students belonging to each of the following groups of marks:

0-9, 10-19, 20-2990-100

OUTPUT SOMETHING LIKE -

GROUP	RANGE	FREQUENCY
1	0-9	2
2	10-19	4
.....		

5# Implement Breadth First Search (BFS) in C/C++.

6# Implement Depth First Search (DFS) in C/C++.

WEEK #12

OBJECTIVES

- To enhance Programming and Software Engineering skills and techniques using SQL and PL/SQL.
- To learn the use of Procedures, Functions and Triggers in database.
- To develop an understanding of the differences between OODBMS, ORDBMS and RDBMS and the practical implications of each approach.
- To analyze and design a real database application.

OUTCOMES

After completing this, the students would be able to:

- understand programming and software engineering skills and techniques using Oracle, SQL and PL/SQL.
- implement procedures, functions and Triggers in database.
- design a real database application.

PROBLEMS

- 1# Create a table of Employee (emp-number, name, dept, salary) and Department (dept number, dept name). Insert some records in the tables through appropriate forms having integrity checks. Also add some records in employee table where department value is not present in department table. Now answer the following query:
- a) Display all records from employee table where department is not found in department table.

- b) Create another table: Sales_data (Region-code, City, Salesperson-code, Sales-qty).
- c) Display records where salesperson has achieved sales more than average sales of all sales persons of all the regions.
- d) Write a procedure/trigger on department code so such that validity of the code is checked and the name of department is automatically displayed on selection of department code. Assume, design and create the necessary tables and constraints.
- e) Write a procedure/trigger on a numeric field named value1 to check if the entered value is 1 (Married) or 2 (Unmarried). In case, the entered value is 1 (Married) then the control should pass to a field named 'spouse name' or else it goes to a field named: Father's Name.
- f) Employee code must begin with 'P' (permanent) or 'T' (Temporary) and its second character must be a digit.
- g) Write procedure /trigger to check if the entered value is correct.

WEEK #13

OBJECTIVES

- To learn how to create an Index on Database.
- To learn how to handle the access privileges in the tables/database.

OUTCOMES

After completing this, the students would be able to:

- Create and handle the Index in database.
- Implement the access privileges on the tables in the database.

PROBLEMS

1# Create following tables with appropriate data within them-

- a. CLIENT_MASTER
- b. PRODUCT_MASTER
- c. SALESMAN_MASTER
- d. SALES_ORDER
- e. SALES_ORDER DETAILS

Indexes-

- i. Create an index on the column 'State' of Client_Master.
- ii. Create a unique index on column 'Pincode' of Product_Master.
- iii. Create a composite unique index on (OrderNo, ProductNo) of Sales_Order_Details.
- iv. Show the rowids of Sales_Order table.

- v. Drop all the above created indexes.

Views-

- i. Create a view from Product_Master.
- ii. Create a view from Salesman_Master table having columns SALESMANNO, SALAMT, CITY, REMARKS and only for salesmen who earn 3000 Rs. or less.
- iii. Create a view 'PS' which has records from both Product_Master and Sales_Order_Details.
- iv. Set 'QtyOrdered' to '5' for 'P07868' in Sales_Order_Details using view 'PS'.
- v. Delete the records of Product_Master for which description is 'Trousers'.

Access Privileges-

- i. Create a role 'Allrole' to perform all operations on Sales_Order_Details.
- ii. Create a user 'Mamta'.
- iii. Give 'Mamta' the permission to view records in Product_Master and Client_Master with further grant permission on these tables to other users.
- iv. Let 'Mamta' to assign the permission to view records in Product_Master and Client_Master to a new user 'Bill'.
- v. Assign role 'Allrole' to a new user 'Ivan'.
- vi. Revoke the assigned permissions from 'Mamta'.
- vii. Delete role 'Allrole'.
- viii. Delete all users.

WEEK #14

OBJECTIVE

- To learn Exception Handling in the database.
- To learn the use of Cursor in the SQL programming.
- To learn the use of Procedures, Functions and Triggers in database.

OUTCOMES

After completing this, the students would be able to:

- Understand and handle the various exceptions that occur in the database.
- Understand and implement the use of Cursor in the SQL programming.
- Write the Procedures, Functions and Triggers in database.

PROBLEMS

1# Create following tables with appropriate data within them-

- i. CLIENT_MASTER
- ii. PRODUCT_MASTER
- iii. SALESMAN_MASTER
- iv. SALES_ORDER
- v. SALES_ORDER DETAILS

Generic PL/SQL Block-

- i. Write a PL/SQL block to multiply two numbers.

- ii. Write a PL/SQL block to find the minimum among three numbers.
- iii. Create a table as follows:

FAC (num number, fac number)

- i. Write a PL/SQL block that accepts a limit from user, calculate factorial of all numbers till that limit and inserts the number and corresponding factorial value in the table FACas given below:

Limit- 5

Num	Fac
1	1
2	2
3	6
4	24
5	120

- i. Write a PL/SQL block that accepts 'ProductNo' from user and displays corresponding 'QtyOrdered' from Sales_Order_details table.
- ii. Write a PL/SQL block that accepts 'OrderNo', 'ProductNo' and 'QtyOrdered' from user and updates corresponding record in Sales_Order_details table only when 'QtyOrdered' is less than or equal to the 'QtyOnHand' value of Product_Master table.

Exception Handling-

- i. Extract record from Client_Master table where state is 'Maharashtra'. If more than one record is extracted then handle the error using 'TOO_MANY_ROWS' exception.
- ii. ORA-02291 is the foreign key violation error number. Handle this exception in a more user-friendly manner. (Consider tables of your choice).
- iii. Calculate the square root of a given number. If the entered number is negative then handle this situation with the help of a user-defined

exception. (SQRT is an inbuilt function in oracle to calculate square root.)

Cursor-

- i. Extract record from Client_Master table where state is 'Maharashtra'.and display corresponding Client_No and Client_Name.
- ii. Read the Client_No from Client_Master table who lives in 'Mumbai' and then change the OrderStatus to 'Fulfilled' for these clients in Sales_Order table.

Function and Procedure-

- i. Write a function/stored procedure which takes a date and displays its day. (Sunday, Monday, etc.)
- ii. Write a function/stored procedure for a simple calculator for two numbers.
- iii. Write a function to check whether a number is prime or not.
- iv. Write a procedure that reads the records from Sales_Order_detail table using cursor andtells the delay between OrderDate and DelyDate.
- v. Drop all functions/procedures.

Triggers-

- i. Write a trigger which does not allow to insert a record in Sales_Order_detail table if QtyOrdered is greater than QtyOnHand column of Product_Master table.
- ii. Write a trigger for Unique Constraint. (Whenever a value is inserted/updated in a column, it checks whether that value already exists in that column or not).

- iii. Add a new column 'CityChange' in Sales_Order table. Write a trigger to achieve that whenever Client's city changes, the corresponding column should be set to 'Y'.
- iv. Drop all triggers.