MRCET CAMPUS

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS INSTITUTION - UGC, GOVT. OF INDIA)

Affiliated to JNTUH; Approved by AICTE, NBA-Tier 1 & NAAC with A-GRADE | ISO 9001:2015

Maisammaguda, Dhulapally, Komaplly, Secunderabad - 500100, Telangana State, India

LABORATORY MANUAL & RECORD

Name:.....

Roll No:.....Branch:.....

Year:.....Sem:.....

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS INSTITUTION - UGC, GOVT. OF INDIA)

Affiliated to JNTUH; Approved by AICTE, NBA-Tier 1 & NAAC with A-GRADE | ISO 9001:2015

Maisammaguda, Dhulapally, Komaplly, Secunderabad - 500100, Telangana State, India

Certificate

Certified that this is the Bonafide Record of the Work Done by

Mr./Ms.....Roll.No.....of

B.Tech.....year..... Semester for Academic year.....

in.....Laboratory.

Date: Faculty Incharge HOD

Internal Examiner External Examiner

INDEX

S.No	Date	Name of the Activity/Experiment	Grade/ Marks	Faculty Signature

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

LAB MANUAL
(R24A0583)

B.TECH

(II YEAR – I SEM)
(2025-26)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence & Machine Learning)

MALLAREDDYCOLLEGEOFENGINEERING&TECHNOLOGY
(AutonomousInstitution–UGC,Govt.ofIndia)

Recognizedunder 2(f)and12(B)ofUGCACT1956
(AffiliatedtoJNTUH,Hyderabad,ApprovedbyAICTE-AccreditedbyNBA&NAAC–'A'Grade-ISO9001:2015Certified)
Maisammaguda, Dhulapally(PostVia.Hakimpet),Secunderabad–500100,TelanganaState,India

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

Vision

To be a premier center for academic excellence and research through innovative interdisciplinary collaborations and making significant contributions to the community, organizations, and society as a whole.

.

Mission

- To impart cutting-edge Artificial Intelligence technology in accordance with industry norms.
- To instil in students a desire to conduct research in order to tackle challenging technical problems for industry by sustaining the ethical values.
- To develop effective graduates who are responsible for their professional growth, leadership qualities and are committed to lifelong learning.

Quality Policy

- To provide sophisticated technical infrastructure and to inspire students toreach their full potential.
- To provide students with a solid academic and research environment for a comprehensive learning experience.
- To provide research development, consulting, testing, and customized training to satisfy specific industrial demands, thereby encouraging self-employment and entrepreneurship among students.

Programme Educational Objectives (PEO):

Graduates of the program will be able to

PEO1: Build successful careers in AI & ML and related fields by applying fundamental concepts of computer science, maths and specialized knowledge of intelligent systems.

PEO2: Design and implement AI-based solutions to real-world problems, demonstrating creativity, critical thinking.

PEO3: Leverage the professional expertise to enter the workforce, seek higher education, and conduct research on AI-based problem resolution.

PEO4: Uphold ethical values and consider societal, legal, and environmental consequences while developing intelligent systems, safeguarding responsible AI development.

Programme Specific Outcomes (PSO):

After successful completion of the program a student is expected to have specific abilities to:

PSO1: Analyze and examine the fundamental issues with AI and ML applications.

PSO2: Apply machine learning, deep learning, and artificial intelligence approaches to address issues in social computing, healthcare, computer vision, language processing, speech recognition, and other domains.

PSO3: Use cutting-edge AI and ML tools and technology to further your study and research.

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

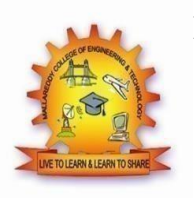
PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

KNOWLEDGE AND ATTITUDE PROFILE (WK)

1. **WK1:** A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
2. **WK2:** Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
3. **WK3:** A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
4. **WK4:** Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
5. **WK5:** Knowledge, including efficient resource use, environmental impacts, whole-life cost, reuse of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
6. **WK6:** Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
7. **WK7:** Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
8. **WK8:** Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
9. **WK9:** Ethics, inclusive behaviour and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.



MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Maisammaguda, Dhulapally Post, Via Hakimpet, Secunderabad – 500100

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:

a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.

b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.

c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab Observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must Maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completingthe task (experiment) in all aspects. He/she must ensure the system / seat is kept properly

HEAD OF THE DEPARTMENT **PRINCIPAL**

Lab Objectives:

- To prepare students to become familiar with the Standard Java technologies ofJ2SE
- To prepare students to excel in Object Oriented programming and to succeed as a Java Developer through global rigorous education.
- To provide Students with a solid foundation in OOP fundamentals required to solve programming problems and also to learn Advanced Java topics like J2ME, J2EE,JSP,JavaScript
- To train Students with good OOP programming breadth so as to comprehend, analyze, design and create novel products and solutions for their al life problems.
- To inculcate in students professional and ethical attitude, multi disciplinary approach and an ability to relate java programming issues to broader application context.
- To provide student with an academic environment aware of excellence, writ ten ethical codes and guidelines and life long learning needed for a successful professional career.

Lab Out comes:

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

- Able to analyze the necessity for Object Oriented Programming paradigm and over structured programming and become familiar with the fund a mental concepts in OOP.
- Demonstrate an ability to design and develop java programs, analyze, and interpret object-oriented data and report results.
- Demonstrate an ability to design an object-oriented system ,AWT components or multi threaded process as per needs and specifications.
- Demonstrate an ability to visualize and work on laboratory and multi disciplinary tasks like console and windows applications both for stand lone and Applets programs

Introduction about lab

System configurations are as follows:

- **Hardware/Software's installed: I**ntel®CORE™i3-3240CPU@3.40GHZRAM:4GB/C,C++Compiler,JAVAJDK1.8,EditPlus.
- Systemsareprovidedforstudentsinthe**1:1ratio.**
- Systemsareassignednumbersandsamesystemisallottedforstudentswhentheydothelab.
- All Systems are configured in LINUX, it is open source and students can use any different programming environments through package installation.

Guidelines to students

A. Standard operating procedure

a) Explanationontoday'sexperimentbytheconcernedfacultyusingPPTcoveringthefollowingaspects:

1) Name of the experiment
2) Aim
3) Software/Hard ware requirements
4) Writing the java programs by the students
5) Commands for executing programs

Writing of the experiment in the Observation Book

The students will write the today's experiment in the Observation book as per the following format:

a) Name of the experiment
b) Aim
c) Writing the program
d) Viva-Voce Questions and Answers
e) Errors observed(if any)during compilation/execution

B. Guide Lines to Students in Lab

Disciplinary to be maintained by the students in the Lab

- Studentsarerequiredtocarrytheirlabobservationbookandrecordbookwithcompletedexperimentswhile enteringthelab.
- Studentsmustusetheequipmentwithcare.Anydamageiscausedstudentispunishable.
- Students are not allowed to use their cell phones/pen drives/CDs in labs.
- Students need to be maintain proper dress code along with ID Card

- Students are supposed to occupy the computers allotted to them and are not supposed to talk or make noise in the lab.
- Students, after completion of each experiment they need to be updated in observation notes and same to be updated in the record.
- Lab records need to be submitted after completion of experiment and get it corrected with the concerned lab faculty.
- If a student is absent for any lab, they need to be completed the same experiment in the free time before attending next lab.

Steps to perform experiments in the lab by the student

Step1: Students have to write the date,aim and for that experiment in the observation book.

Step2: Students have to listen and understand the experiment explained by the faculty and note down the important points in the observation book.

Step3: Students need to write procedure/algorithm in the observation book.

Step4: Analyze and Develop/implement the logic of the program by the student in respective platform

Step5: After approval of logic of the experiment by the faculty then the experiment has to be executed on the system.

Step6: After successful execution the results are to be shown to the faculty and noted the same in the observation book.

Step7: Students need to attend theViva-Voce on that experiment and write the same in the observation book.

Step8: Update the completed experiment in the record and submit to the concerned faculty in-charge.

Instructions to maintain the record

- Before start of the first lab they have to buy the record and bring the record to the lab.
- Regularly (Weekly) update the record after completion of the experiment and get it corrected with concerned lab in-charge for continuous evaluation. In case the record is lost inform the same day to the faculty in charge and get the new record within 2 days the record has to be submitted and get it corrected by the faculty.
- If record is not submitted in time or record is not written properly, the evaluation marks (5M)will be deducted.

Awarding the marks for day to day evaluation

Total marks for day to day evaluation is 15 Marks as per Autonomous (JNTUH). These 15 Marks are distributed as:

Regularity	3Marks
Program written	3Marks
Execution & Result	3Marks
Viva-Voce	3Marks
Dress Code	3Marks

Allocation of Marks for Lab Internal

Total marks for lab internal are 40 Marks as per Autonomous (JNTUH.)

These 40 Marks are distributed as:

Average of day to day evaluation marks:15 Marks

Lab Mid exam:40 Marks

VIVA&Observation:10 Marks

Allocation of Marks for Lab External

Total marks for lab Internal and External are 60Marks as per Autonomous/ (JNTUH).

These 60 External Lab Marks are distributed as:

Program Written	15Marks
Program Execution and Result	25Marks
Viva-Voce	10Marks
Record	10Marks

C. General laboratory instructions

1. Students are advised to come to the laboratory at least 5minutes before (to the starting time), those who come after 5minutes will not be allowed in to the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis/ program/experiment details.
3. Student should enter in to the laboratory with:

a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.

b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.

c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with so phisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attracts evere punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if any body found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seats kept properly.

Head of the Department **Principal**

INDEX

WEEK-1 Date:

Aim:a)Write a java program to find the Fibonacciseries using recursive and non-recursive functions

Program:

```
classfib
{
inta,b,c;
voidnonrecursive(intn)                    //Nonrecursive functiontofindtheFibonacciseries.
{a=0
;b=1
;
System.out.print(a+ "" +
b);c=a+b;
while(c<=n)
{
System.out.print(c);a
=b;
b=c;c=
a+b;
}
}
intrecursive(intn)                    //RecursivefunctiontofindtheFibonacciseries.
{
if(n==0)ret
urn
(0);if(n==1
)return
(1);else
 return(recursive(n-1)+recursive(n-2));
}
}
                                    //Classthat callsrecursiveand nonrecursive functions
classfib1
{
Public staticvoid  main(Stringargs[])
{
intn=5;
System.out.println("TheFibonacciseriesusingnonrecursiveis");
                                    //Creatingobjectforthefib class.
fibf=new fib();
                                    //Callingnon recursive functionoFfibclass.
f.nonrecursive(n);
System.out.println("\n The Fibonacci series using recursive
is");for(inti=0;i<=n;i++)
{
                                    //Callingrecursivefunction offibclass.
int F1=f.recursive(i);System.
out. print(F1);
}
}
}
```

Three Test Outputs:

Output 1:

b)Write a program to multiply two given matrices.

```
public class MatrixMultiplication{
public static void main(String args[]){
//creating two matrices
int a[][]={{2,2,2},{4,4,4},{3,3,3}};
int b[][]={{1,1,1},{2,2,2},{3,3,3}};

//creating another matrix to store the multiplication of two matrices
int c[][]=new int[3][3];  //3 rows and 3 columns

//multiplying and printing multiplication of 2 matrices
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=0;
for(int k=0;k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
}//end of k loop
System.out.print(c[i][j]+" ");  //printing matrix element
}//end of j loop
System.out.println();//new line
}
    }}
```

ThreeTestOutputs:

Output 1:

c) Write a program for Method overloading and Constructor overloading

```
importjava.io.*;
 classMethodOverloadingEx{

   staticint add(int a, int b)
   {
    returna +b;
   }

   staticint add(int a, int b, intc)
   {
      returna +b +c;
   }

   publicstatic void main(Stringargs[])
   {
        System.out.println("add()with2parameters");System.out.println(ad
        d(4,6));

        System.out.println("add() with 3
      parameters");System.out.println(add(4,6, 7));
   }}
```

Output:

Signature of the faculty

EXERCISE:

1. Write a java program to print the multiplication table.
2. Write a java program to find the Factorial of a given integer using recursive and non recursive functions
3. Write a java program that prompts the user for an integer and then print outs all prime numbers up to that integer.

WEEK-2 Date:

Aim:Write a program to demonstrate execution of static blocks ,static variables & static methods.

```
public class StaticBlock {
   static int x = 20;                    //Static variable
   static int y;
   static void func(int z) {             //Static method
System.out.println("x = " + x);
System.out.println("y = " + y);
System.out.println("z = " + z);
   }
   static {                  // Static block
System.out.println("Running static initialization block.");
      y = x + 10;
   }
   public static void main(String args[]) {
func(10);
   }
}
```

Output:

b) Write a program to display the employee details using Scanner class

```
import java.util.Scanner;
class Employee
{
int id;
String name;
String desig;
float salary;
}
class Main{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.print("How many employees? ");
int n = sc.nextInt();
Employee emp[] = new Employee[n];
for (int i = 0; i< n; i++) {
emp[i] = new Employee();


System.out.println("Enter " + (i + 1) + " Employee data :");
System.out.print("Enter employee id :");
emp[i].id = sc.nextInt();
System.out.print("Enter employee name :");
emp[i].name = sc.next();
System.out.print("Enter employee designation :");
emp[i].desig = sc.next();
System.out.print("Enter employee salary :");
emp[i].salary = sc.nextFloat();
}
System.out.println("\n\n********* All Employee Details are :*********\n");
for (int i = 0; i< n; i++) {
System.out.println("Employee id, Name, Designation and Salary :"
+ emp[0].id + " " + emp[i].name + " " + emp[i].desig + " " +emp[i].salary);
}
}
}
```

Output:

a) Write a program for sorting a given list of names in ascending order

```
import java.util.Scanner;
public class Alphabetical_Order
{
   public static void main(String[] args)
   {
      int n;
      String temp;
      Scanner s = new Scanner(System.in);
System.out.print("Enter number of names you want to enter:");
      n = s.nextInt();
      String names[] = new String[n];
      Scanner s1 = new Scanner(System.in);
System.out.println("Enter all the names:");
for(int i = 0; i< n; i++)
      {
         names[i] = s1.nextLine();
      }
      for (int i = 0; i< n; i++)
      {
         for (int j = i + 1; j < n; j++)
         {
            if (names[i].compareTo(names[j])>0)
            {
               temp = names[i];
               names[i] = names[j];
               names[j] = temp;
            }
         }
      }
System.out.print("Names in Sorted Order:");
      for (int i = 0; i< n - 1; i++)
      {
System.out.print(names[i] + ",");
      }
System.out.print(names[n - 1]);
   }
}
```

Output:

Signature of the faculty

EXERCISE:

1. Write a java program to transpose of a matrix is obtained by changing rows to cols & columns to rows.
2. Write a java program to check whether the Matrix is Symmetric or Not.
3. Write a Java Program to find Matrix is an Identity Matrix or not.
4. Write a java program to add and subtract two given matrices.

WEEK-3 Date:

Aim: a) Write a program to implement single and Multi level inheritance

```
class Animal{
void eat(){
System.out.println("eating...");
} }
class Dog extends Animal{
void bark(){
System.out.println("barking...");
} }
class BabyDog extends Dog{
void weep(){
System.out.println("weeping...");
} }
class Multilevel{
public static void main(String args[]){
BabyDog d=new BabyDog();
d.weep();
d.bark();
d.eat();
}}
```

Output:

b) Write a program to implement Hierarchical Inheritance.

```
class Animal{
void eat(){
System.out.println("eating...");
}}
class Dog extends Animal{
void bark(){
System.out.println("barking...");
}}
class Cat extends Animal{
void meow(){
System.out.println("meowing...");
}}
class Hierarchical{
public static void main(String args[]){
Cat c=new Cat();
c.meow();
c.eat();
}}
```

Output:

c)Write a program to implement method overriding

```
//Creating a parent class.
class Vehicle{
  //defining a method
  void run(){
System.out.println("Vehicle is running");
}  }
//Creating a child class
class Bike extends Vehicle{
  //defining the same method as in the parent class
  void run(){
System.out.println("Bike is running safely");
}
    public static void main(String args[]){
Bike obj = new Bike();//creating object
  obj.run();//calling method
  }  }
```

Output:

Signature of the faculty

EXERCISE:

1. Write a java program to find all even and odd integers up to a given integer.
2. Write a java program that reads a line of integers and displays each integers and the product of all integers uses String Tokenize.

WEEK-4 **Date:**

a) Write a program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void area(double x,double y);
}
class Rectangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of triangle is :"+(0.5*x*y));
}
}
public class AbstactDDemo
{
public static void main(String[] args)
{
Rectangle r=new Rectangle();
r.area(2,5);
Circle c=new Circle();
c.area(5,5);
Triangle t=new Triangle();
t.area(2,5);
}
}
```

Output:

b) Write a program to implement Interface

```
interface Drawable{
void draw();
}
//Implementation: by second user
class Rectangle implements Drawable{
public void draw(){
System.out.println("drawing rectangle");
}  }
class Circle implements Drawable{
public void draw(){
System.out.println("drawing circle");
}  }
//Using interface: by third user
class Interface{
public static void main(String args[]){
Drawable d=new Circle();//In real scenario, object is provided by method e.g. getDrawable()
d.draw();
}}
```

Output:

b) Write a program to implement constructor overloading

```
// Java program to illustrate
// Constructor Overloading
class Box {
	double width, height, depth;

	// constructor used when all dimensions
	// specified
	Box(double w, double h, double d)
	{
		width = w;
		height = h;
		depth = d;
	}

	// constructor used when no dimensions
	// specified
	Box() { width = height = depth = 0; }

	// constructor used when cube is created
	Box(double len) { width = height = depth = len; }

	// compute and return volume
	double volume() { return width * height * depth; }
}

// Driver code
public class constructor Overloading {
	public static void main(String args[])
	{
		// create boxes using the various
		// constructors
		Box mybox1 = new Box(10, 20, 15);
		Box mybox2 = new Box();
		Box mycube = new Box(7);

		double vol;

		// get volume of first box
		vol = mybox1.volume();
		System.out.println("Volume of mybox1 is " + vol);

		// get volume of second box
		vol = mybox2.volume();
		System.out.println("Volume of mybox2 is " + vol);

		// get volume of cube
		vol = mycube.volume();
		System.out.println("Volume of mycube is " + vol);
```

```
        }
}
```

Output:

Signature of the faculty

EXERCISE:

1. Write a java program to Read and display the student details using Scanner class.
2. Write a java program that displays the number of characters, lines, words, white spaces in a text file.

WEEK-5 Date:

Aim:a)Write a program to create inner classes

```
class Outer_place
{
int num;
private class Inner_place
{
public void print()
{
System.out.println("It is an inner class");
}
}
void display_Inner()
{
Inner_place inner = new Inner_place();
inner.print();
}
}
public class My_class
{
public static void main(String args[])
{
Outer_place outer = new Outer_place();
outer.display_Inner();
}
}
```

Output:

b)Write a program to create user defined package and demonstrate various access modifiers.

A.javapackag
e
pack;publiccl
assA
{
publicvoidmsg()
{
System.out.println("Hello");
}
}

B.java
import
pack.A;classB
{
publicstaticvoid main(Stringargs[])
{
A obj = new
A();obj.msg();
}
}

Output:

Demonstration of various access modifiers:

private access modifier:

```
class A{
private int data=40;
private void msg(){System.out.println("Hello java");}
}

public class Simple{
 public static void main(String args[]){
   A obj=new A();
   System.out.println(obj.data);
   obj.msg();
   }  }
```

Output:

Protected Access Modifier:

```
class Animal {
   // protected method
   protected void display() {
System.out.println("I am an animal");
   }}
class Dog extends Animal {
   public static void main(String[] args) {
      // create an object of Dog class
      Dog dog = new Dog();
       // access protected method
dog.display();
   }
}
```

Output:

Public Access Modifier:

```
// public class
public class Animal {
   // public variable
   public int legCount;
   // public method
   public void display() {
System.out.println("I am an animal.");
System.out.println("I have " + legCount + " legs.");
   }
}
public class PublicA {
   public static void main( String[] args ) {
      // accessing the public class
      Animal animal = new Animal();
      // accessing the public variable
animal.legCount = 4;
      // accessing the public method
animal.display();
   }
}
```

Output:

Default Access Modifier:

if we do not explicitly specify any access modifier for classes, methods, variables, etc, then by default the default access modifier is considered

`

C) Write a program to demonstrate the use of super and final keywords

Super:

```
class employee {
 int wt = 8;
}

class clerk extends employee {
 int wt = 10;  //work time
 void display() {
System.out.println(super.wt); //will print work time of clerk
 }

 public static void main(String args[]) {
  clerk c = new clerk();
c.display();
 }
}
```

Output:

Final key Word:

```
final class stud {
Void disp() {
System.out.println("Final method");

}

}
class books extends stud {
 void show() {
System.out.println("Book-Class method");
 }
 public static void main(String args[]) {
  books B1 = new books();
  B1.show();
B1.disp();
 }
}
```

Output:

Signature of the faculty

EXERCISE:

1. Write a java program to sort the given integers in ascending/descending order.
2. Write a java program to display characters in a string in sorted order.
3. Write a program that uses a sequence in put stream to output the contents of two files.

WEEK– 6A Date:

a) Write a program if number is less than 10 and greater than 50 it generate the exception out of range. Else it displays the square of number

```
class OutOfRangeException extends Exception {
   public OutOfRangeException(String message) {
      super(message);
   }
}

public class NumberProcessor {
   public static void main(String[] args) {
      try {
         int userInput = getUserInput();
processNumber(userInput);
      } catch (NumberFormatException e) {
System.out.println("Invalid input. Please enter a valid number.");
      } catch (OutOfRangeException e) {
System.out.println("Error: " + e.getMessage());
      }
   }

   private static int getUserInput() throws NumberFormatException {
System.out.print("Enter a number: ");
      String input = System.console().readLine();
      return Integer.parseInt(input);
   }

   private static void processNumber(int number) throws OutOfRangeException {
      if (number < 10 || number > 50) {
         throw new OutOfRangeException("Number is out of range (less than 10 or greater than 50)");
      } else {
         int result = number * number;
System.out.println("The square of " + number + " is: " + result);
      }
   }
}
```

ThreeTestOutputs:

Signature of the faculty

b) Write a program with multiple catch statements.

```
public class MultipleCatchBlock1 {

   public static void main(String[] args) {

try{
            int a[]=new int[5];
a[5]=30/0;
           }
catch(ArithmeticException e)
            {
System.out.println("Arithmetic Exception occurs");
            }
catch(ArrayIndexOutOfBoundsException e)
            {
System.out.println("ArrayIndexOutOfBounds Exception occurs");
            }
catch(Exception e)
            {
System.out.println("Parent Exception occurs");
            }
System.out.println("rest of the code");
   }
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 6C Date:

Write a program to implement nested try

```
public class NestedTryBlock
   {
   public static void main(String args[]){
   //outer try block
            try
                     {
                     //inner try block 1
                     try
                      {
                     System.out.println("going to divide by 0");
                        int b =39/0;
                      }
            //catch block of inner try block 1
catch(ArithmeticException e)
                     {
                              System.out.println(e);
                     }


//inner try block 2
            try{
                     int a[]=new int[5];

             //assigning the value out of array bounds
                     a[5]=4;
             }

       //catch block of inner try block 2
            catch(ArrayIndexOutOfBoundsException e)
            {
                     System.out.println(e);
            }


            System.out.println("other statement");
             }
            //catch block of outer try block
            catch(Exception e)
            {
                     System.out.println("handled the exception (outer catch)");
            }

            System.out.println("normal flow..");
            } }
```

ThreeTestOutputs:

Output 1:

Signature of the faculty

WEEK– 7A Date:

Write a Program to implement simple Thread by extending Thread class and implementing runnable interface.

```
class Multi extends Thread{

public void run(){

System.out.println("thread is running...");

}

public static void main(String args[]){

Multi t1=new Multi();

t1.start();

 }

}
```

ThreeTestOutputs:

Signature of the faculty

```
class Multi3 implements Runnable
{
public void run()
{
System.out.println("thread is running...");
}

public static void main(String args[]){
Multi3 m1=new Multi3();
Thread t1 =new Thread(m1);   // Using the constructor Thread(Runnable r)
t1.start();
}  }
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 7B Date:

Write a program that implements a multi-thread application that has three threads

```
classThread1extendsThread
{
publicvoidrun()
{
for(inti=0;i<=5;i++)
{
System.out.println("Thread1:"+i);
}
}
}
classThread2extendsThread
{
publicvoidrun()
{
for(intj=0;j<=5;j++)
{
System.out.println("Thread2:"+j);
}
}
}
classThread3extendsThread
{
publicvoidrun()
{
for(intk=0;k<=5;k++)
{
System.out.println("Thread3:"+k);
} } }
classMultiThreadDemo
{
publicstaticvoidmain(Stringargs[])
{
Thread1
        t1=newTh
read1();
Thread2t2=new
Thread2();
Thread3
        t3=newTh
read3();
t1.start();
```

```
t2.start();
t3.start();

for(inti=0;i<=5;i++)
{
System.out.println("mainthread:"+i);
}
} }
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 7C Date:

Write a program to set and print thread priorities

```
public class JavaSetPriorityExp1 extends Thread
{
public void run()
{
System.out.println("Priority of thread is: "+Thread.currentThread().getPriority());
}
public static void main(String args[])
{
 // creating one thread
 JavaSetPriorityExp1 t1=new JavaSetPriorityExp1();
 // print the maximum priority of this thread
 t1.setPriority(Thread.MAX_PRIORITY);
 // call the run() method
 t1.start();
}
}
```

ThreeTestOutputs:

Signature of the faculty

Example 2: Minimum Priority Thread

```
public class JavaSetPriorityExp2 extends Thread
{
   public void run()
   {
System.out.println("Priority of thread is: "+Thread.currentThread().getPriority());
   }
   public static void main(String args[])
   {
      // creating one thread
      JavaSetPriorityExp2 t1=new JavaSetPriorityExp2();
      // print the minimum priority of this thread
      t1.setPriority(Thread.MIN_PRIORITY);
      // This will call the run() method
      t1.start();
   }
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 8

8: Write a program to implement following collections a)array List b) Vector c)Hash table d)Stack

a) Date:

```
importjava.util.*;
classTestJavaCollection
{
publicstaticvoidmain(Stringargs[])
{
ArrayList<String>list=newArrayList<String>();          //Creating
arraylistlist.add("Kalpana");                  //Adding object in
arraylistlist.add("Venu");
list.add("Suneetha");
list.add("Gayatri");
        //Traversing list through
IteratorIteratoritr=list.iterator();while(it
r.hasNext())
{
  System.out.println(itr.next());
}
}
}
```

Test output:

Signature of the faculty

b)Vecor

```
import java.util.*;
public class VectorExample {
      public static void main(String args[]) {
        //Create a vector
        Vector<String>vec = new Vector<String>();
        //Adding elements using add() method of List
vec.add("Tiger");
vec.add("Lion");
vec.add("Dog");
vec.add("Elephant");
        //Adding elements using addElement() method of Vector
vec.addElement("Rat");
vec.addElement("Cat");
vec.addElement("Deer");

System.out.println("Elements are: "+vec);
      }
}
```

c) Hash table

```
import java.util.*;
public class HashTableDemo {

  public static void main(String args[]) {
    // Create a hash map
Hashtable balance = new Hashtable();
    Enumeration names;
    String str;
    double bal;

balance.put("Jack", new Double(400.52));
balance.put("peter", new Double(500.88));
```

```
balance.put("Mark", new Double(866.68));
balance.put("James", new Double(57.69));
balance.put("Qulit", new Double(36.33));

     // Show all balances in hash table.

    names = balance.keys();

     while(names.hasMoreElements()) {
       str = (String) names.nextElement();
System.out.println(str + ": " + balance.get(str));
     }
System.out.println();

     // Deposit 200 into jack's account
bal = ((Double)balance.get("Jack")).doubleValue();
balance.put("Jack", new Double(bal + 1000));
System.out.println("Jack new balance: " + balance.get("Jack"));
  }
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 9 Date:

A)write a Java program for addition of two numbers using lambda expression

```
interface Addable{
   int add(int a,int b);
}

public class LambdaExpression {
   public static void main(String[] args) {

      // Lambda expression without return keyword.
      Addable ad1=(a,b)->(a+b);
System.out.println(ad1.add(10,20));

      // Lambda expression with return keyword.
      Addable ad2=(int a,int b)->{
                     return (a+b);
                     };
System.out.println(ad2.add(100,200));
   }
}
```

Threetestoutputs:

Signature of the faculty

B) Write a java program for producer and consumer problem using Threads

```
Class Inter Thread Demo
{
publicstaticvoid main(Stringargs[])
{
Producer p1=new
Producer();Consumer c1=new
Consumer(p1);Thread t1=new
Thread(p1);Thread t2=new
Thread(c1);t2.start();
t1.start();
}
}
classProducerextendsThread
{
StringBuffersb;
Producer()
{
sb=newStringBuffer();
}
publicvoid run()
{
synchronized(sb)
{
for(inti=0;i<=10;i++)
{
try
{
sb.append(i+":");Thread.sleep(10
00);System.out.println("appendi
ng");
}
catch(InterruptedExceptione)
{
System.out.println(e);
}
}
sb.notify();
}
}
}
```

```
classConsumerextendsThread
{
Producerprod;
Consumer(Producerprod)
{
this.prod=prod;
}
publicvoid run()
{
synchronized(prod.sb)
{
try
{
prod.sb.wait();
}
catch(Exceptione)
{
System.out.println(e);
}
System.out.println(prod.sb);
}
}
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 10 Date:

a)Write a program to list all the files in a directory including the files present in all its subdirectories.

```
// Java program to print all files

import java.io.File;

public class GFG {
static void RecursivePrint(File[] arr, int index, int level)
{
	// terminate condition
	if (index == arr.length)
		return;

	// tabs for internal levels
	for (int i = 0; i< level; i++)
		System.out.print("\t");

	// for files
	if (arr[index].isFile())
		System.out.println(arr[index].getName());

	// for sub-directories
	else if (arr[index].isDirectory()) {
		System.out.println("[" + arr[index].getName()+ "]");

		// recursion for sub-directories
		RecursivePrint(arr[index].listFiles(), 0,
					level + 1);
	}

	// recursion for main directory
	RecursivePrint(arr, ++index, level);
}

// Driver Method
public static void main(String[] args)
{
	// Provide full path for directory(change
	// accordingly)
	String maindirpath= "C:\\Users\\mrcet\\Desktop\\Test";

	// File object
	File maindir = new File(maindirpath);

	if (maindir.exists() &&maindir.isDirectory()) {

		// array for files and sub-directories
		// of directory pointed by maindir
		File arr[] = maindir.listFiles();

		System.out.println(
			"**********************************************");
```

```
                System.out.println(
                        "Files from main directory : " + maindir);
                System.out.println(
                        "*****************************************");

                // Calling recursive method
                RecursivePrint(arr, 0, 0);
        }
}
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK–10B

Write a Program to Read the Content of a File Line by Line

```
import java.io.BufferedInputStream;
import java.io.FileInputStream;

class Main {
public static void main(String[] args) {
try {

 // Creates a FileInputStream
FileInputStream file = new FileInputStream("input.txt");

 // Creates a BufferedInputStream
BufferedInputStream input = new BufferedInputStream(file);

  // Reads first byte from file
int i = input .read();

 while (i != -1) {
System.out.print((char) i);

  // Reads next byte from the file
i = input.read();
 }
input.close();
 }

 catch (Exception e) {
e.getStackTrace();
  }
}
}
```

ThreeTestOutputs:

Signature of the faculty

WEEK– 11 Date:

Write a program that connects to a database using JDBC display all records in a table.

Create Connection:

```
import java.sql.*;

public class connection {

    Connection con = null;

    public static Connection connectDB()

    {

      try {
            // Importing and registering drivers
            Class.forName("com.mysql.jdbc.Driver");

            Connection con = DriverManager.getConnection(
                  "jdbc:mysql://localhost:3306/mydb",
                  "root", "1234");
            // here,root is the username and 1234 is the
            // password,you can set your own username and
            // password.
            return con;
      }
      catch (SQLException e) {

            System.out.println(e);
      }
    }
}
```

Fetching data from database table:

```
import java.sql.*;

public class GFG {

    // Step1: Main driver method
    public static void main(String[] args)
    {
```

```
    // Step 2: Making connection using
    // Connection type and inbuilt function on
    Connection con = null;
    PreparedStatement p = null;
    ResultSetrs = null;

    con = connection.connectDB();

    // Try block to catch exception/s
    try {

            // SQL command data stored in String datatype
            String sql = "select * from cuslogin";
            p = con.prepareStatement(sql);
            rs = p.executeQuery();

            // Printing ID, name, email of customers
            // of the SQL command above
            System.out.println("id\t\tname\t\temail");

            // Condition check
            while (rs.next()) {

                    int id = rs.getInt("id");
                    String name = rs.getString("name");
                    String email = rs.getString("email");
                    System.out.println(id + "\t\t" + name+ "\t\t" + email);
            }
    }

    // Catch block to handle exception
    catch (SQLException e) {

            // Print exception pop-up on screen
            System.out.println(e);
    }
}
}
```

Threetestoutputs:

Signature of the faculty

WEEK11B

AimWrite a program to connect to a database using JDBC and insert values into it.

```
import
java.sql.Connection;import
java.sql.DriverManager;import
java.sql.Statement;publicclassP
ostgreSQLJDBC
{
publicstaticvoid main(Stringargs[])
{
Connection c = null; Statement stmt = null;try
{Class.forName("org.postgresql.Driver");c=
DriverManager
  .getConnection("jdbc:postgresql://localhost:5432/testdb", "manisha",
"123");c.setAutoCommit(false);System.out.println("Openeddatabasesuccessfully");
stmt=c.createStatement();
Stringsql="INSERTINTOCOMPANY(ID,NAME,AGE,ADDRESS,SALARY)"
   +"VALUES(1,'Paul', 32,'California',20000.00);";
stmt.executeUpdate(sql);

sql="INSERTINTOCOMPANY(ID,NAME,AGE,ADDRESS,SALARY)"
    +"VALUES(2,'Allen',25,'Texas', 15000.00);";
stmt.executeUpdate(sql);

sql="INSERTINTOCOMPANY(ID,NAME,AGE,ADDRESS,SALARY)"
     +"VALUES (3, 'Teddy',23,'Norway', 20000.00);";
stmt.executeUpdate(sql);

sql="INSERTINTOCOMPANY(ID,NAME,AGE,ADDRESS,SALARY)"
     +"VALUES(4, 'Mark',25,'Rich-Mond', 65000.00);";
stmt.executeUpdate(sql);

stmt.close();
c.commit();
c.close();
}
catch(Exceptione)
{
System.err.println(e.getClass().getName()+":"+e.getMessage());S
ystem.exit(0);
}
System.out.println("Recordscreatedsuccessfully");
}
}
```

testoutputs:

Signature of the faculty

WEEK– 11C Date:

Write a program to connect to a database using JDBC and delete values from it

```
import
java.sql.Connection;import
java.sql.DriverManager;import
java.sql.ResultSet;importjava.s
ql.Statement;

   publicclassPostgreSQLJDBC6{publicstaticvoidmain(Stringargs[])
{
Connection c =
null;Statementstmt=n
ull;try{
Class.forName("org.postgresql.Driver");
c= DriverManager.getConnection
                ("jdbc:postgresql://localhost:5432/testdb", "manisha",
"123");c.setAutoCommit(false);
System.out.println("Openeddatabasesuccessfully");

stmt=c.createStatement();
Stringsql="DELETEfromCOMPANYwhereID=2;";stmt
.executeUpdate(sql);
c.commit();
ResultSetrs=stmt.executeQuery("SELECT*FROMCOMPANY;");whi
 le(rs.next( ))
{
intid=rs.getInt("id");
Stringname=rs.getString("name");in
tage=rs.getInt("age");
String address =
rs.getString("address");float salary =
rs.getFloat("salary");System.out.println(
"ID = " + id
);System.out.println("NAME="+name);Syst
em.out.println("AGE = "+age);
System.out.println("ADDRESS="+address);Syste
m.out.println( "SALARY = " + salary
);System.out.println();
}
rs.close();
stmt.close();
c.close();
}
```

```
catch(Exceptione)
{
System.err.println(e.getClass().getName()+":"+e.getMessage());S
ystem.exit(0);
}
System.out.println("Operationdonesuccessfully");
}
}
```

Threetestoutputs:

Signature of the faculty

WEEK–12

Date:

Writeajavaprogramthatworksasasimplecalculator.UseaGridLayouttoarrangeButtonsfo rdigits andforthe+ -* %operations.Add atextfieldtodisplaytheresult

```
import
javax.swing.*;import
javax.swing.event.*;importj
ava.awt.*;
importjava.awt.event.*;
classAextendsJFrameimplementsActionListener
{
public JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15,
b16;publicJTextField tf1;
publicJPanelp;public
String v = "";public
String v1 = "0";public
String op =
"";publicA()
{
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);setSize(
400,400);
p=newJPanel(newFlowLayout());tf
1 = new JTextField(10);p.add(tf1);
add(p);
setLayout(new GridLayout(0,
3));b1 = new
JButton("1");b1.addActionListen
er(this);add(b1);
b2 = new
JButton("2");b2.addAction
Listener(this);add(b2);
b3 = new
JButton("3");b3.addAction
Listener(this);add(b3);
b4 = new
JButton("4");b4.addAction
Listener(this);add(b4);
```

```
        b5 = new
JButton("5");b5.addAction
```

```
Listener(this);add(b5);
b6 = new
JButton("6");b6.addAction
Listener(this);add(b6);
b7 = new
JButton("7");b7.addAction
Listener(this);add(b7);
b8 = new
JButton("8");b8.addAction
Listener(this);add(b8);
b9 = new
JButton("9");b9.addAction
Listener(this);add(b9);
b10 = new
JButton("0");b10.addAction
Listener(this);add(b10);
b11 = new
JButton("+");b11.addAction
Listener(this);add(b11);
b12 = new JButton("-
");b12.addActionListener(thi
s);add(b12);
b13 = new
JButton("*");b13.addAction
Listener(this);add(b13);
b14 = new
JButton("/");b14.addAction
Listener(this);add(b14);
b16 = new
JButton("%");b16.addAction
Listener(this);add(b16);
b15 = new
JButton("=");b15.addAction
Listener(this);add(b15);
setVisible(true);
publicvoidactionPerformed(ActionEventae)
{
String b =
ae.getActionCommand();switch(b)
{
case"1":
{
v=v+ "1";
```

```
tf1.setText(v);
}
break;case
"2":{
v=v+ "2";
tf1.setText(v);
}
break;case
"3":{
v=v+ "3";
tf1.setText(v);
}
break;case
"4":{
v=v+ "4";
tf1.setText(v);
}
break;case
"5":{
v=v+ "5";
tf1.setText(v);
}
break;case
"6":{
v=v+ "6";
tf1.setText(v);
}
break;


case"7":{
v=v+ "7";
tf1.setText(v);
}
break;case
"8":{
v=v+ "8";
tf1.setText(v);
}
break;case
"9":{
v=v+ "9";
tf1.setText(v);
```

```
}
break;case
"0":{
v=v+ "0";
tf1.setText(v);
}
break;case
"+":{
op="+";
v1 =
tf1.getText();v=""
;
}
break;cas
e"-":{
op="-";
v1 =
tf1.getText();v=""
;
}
break;case
"*":{
op="*";
v1 =
tf1.getText();v=""
;
}
break;cas
e"/":{
op="/";
v1 =
tf1.getText();v=""
;
}
break;
case"%":{
op="%";
v1 =
tf1.getText();v=""
;
}
break;case
"=":{
```

```
switch (op)
{case "+":{
v                   =
tf1.getText();if
(v.equals(""))
{v="0";
}
long i = Long.parseLong(v1) +
Long.parseLong(v);tf1.setText(String.valueOf(i));
v="";
}
break;cas
e"-":{
v                   =
tf1.getText();if
(v.equals(""))
{v="0";
}
long i = Long.parseLong(v1) -
Long.parseLong(v);tf1.setText(String.valueOf(i));
v="";
}
break;
case"*":{
v                   =
tf1.getText();if
(v.equals(""))
{v="0";
}
long i = Long.parseLong(v1) *
Long.parseLong(v);tf1.setText(String.valueOf(i));
v="";
}
break;cas
e "/":
{try{
v                   =
tf1.getText();if
(v.equals(""))
{v="0";
}
long i = Long.parseLong(v1) /
Long.parseLong(v);tf1.setText(String.valueOf(i));
```

```
v="";
} catch (Exception ex)
{JOptionPane.showMessageDialog(this,ex.getMessage())
;
}
}
break;
case "%":
{try{
v                    =
tf1.getText();if
(v.equals(""))
{v="0";
}
longi=Long.parseLong(v1)%Long.parseLong(v);tf1
.setText(String.valueOf(i));
v="";
} catch (Exception ex)
{JOptionPane.showMessageDialog(this,ex.getMessage())
;
}
}
break;
}
}
break;
}
}
}
publicclassCalc
{
publicstaticvoidmain(String[]args)
{
Aa=new A();
}
}
```

Testoutputs: