**PRACTICAL 1**

Q1) Write a java program that takes a number as input and print its multiplication table

        upto 10.

>>> import java.util.Scanner;

public class P1MUL

{

public static void main(String args [])

{

Scanner s=new Scanner(System.in);

System.out.println("Enter a number");

int n=s.nextInt();

for(int i=1;i<=10;i++)

{

System.out.println(n+"\*"+i+"="+(n\*i));

}

}

}

A computer screen shot of a program

AI-generated content may be incorrect.

Q2) Write a program to calculate the area and perimeter of a circle whose radius is taken

        from the user

>>> import java.util.Scanner;

public class P1CIRCLE

{

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

System.out.println("Enter the radius");

double r=s.nextDouble();

double perimeter=2\*3.14\*r;

double area=3.14\*r\*r;

System.out.println("The perimeter is "+perimeter);

System.out.println("The area is "+area);

}

}

A computer screen with white text

AI-generated content may be incorrect.

Q3) A. Write a java to program to demonstrate Command line arguments (of type String)

>>> class P1STRCL

{

public static void main(String args[])

{

System.out.println("Argument count: "+args.length);

for (int i=0;i<args.length;i++)

{

System.out.println("Java is:"+args[i]);

}

}

}

A computer screen shot of a code

AI-generated content may be incorrect.

        B. Write a java to program to demonstrate Command line arguments (of type integer)

>>>> class P1INTCL

{

public static void main(String args[])

{

System.out.println("Argument count:"+args.length);

int sum=0;

for(int i=0;i<args.length;i++)

{

int num=Integer.parseInt(args[i]);

System.out.println("num is "+i+":"+args[i]);

sum=sum+num;

}

System.out.println("Sum is "+sum);

}

}

A screen shot of a computer

AI-generated content may be incorrect.

**Practical No 2**

1. Write a java program to reverse a string taken from the user

# CODE

import java.util.Scanner; class StrRev1

{

public static void main(String args[])

{

String s1,s2=" ";

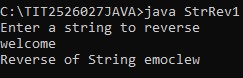
Scanner s=new Scanner(System.in); System.out.println("Enter a string to reverse"); s1=s.nextLine(); int len=s1.length(); for(int i=len-1;i>=0;i--) s2=s2+s1.charAt(i);

System.out.println("Reverse of String"+s2);

}

}

# OUTPUT



2. Write a java program to perform the following using string functions

i) Take two strings from user ii) Display the 5th character of String1 iii) Check if two strings are equal iv) Display the length of String 2

v) Display String 1 in upper case vi) Display String 2 in lower case

# CODE

import java.util.Scanner; class StrFun

{

public static void main(String args[])

{

String s1,s2=" ";

Scanner s=new Scanner(System.in);

System.out.println("Enter first String");

s1=s.nextLine();

System.out.println("Enter second String");

s2=s.nextLine(); char c=s1.charAt(4);

System.out.println("The 5th Char of String 1 is"+c); if(s1.equals(s2))

System.out.println("Strings are equal"); else

System.out.println("Strings are not equal"); int len=s2.length();

System.out.println("Length of String 2 is"+len);

System.out.println("Upper case of String 1"+s1.toUpperCase());

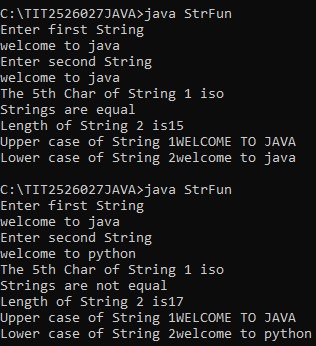
String s3=s2.toLowerCase();

System.out.println("Lower case of String 2"+s3);

}

}

# OUTPUT



3. Create a class Student which has roll no, name and marks of 5 subjects as variables and a method to calculate and print percentage.

# CODE

class Student

{ int rno; String name; float m1,m2,m3,m4,m5; public void calculate(float m1,float m2,float m3,float m4,float m5)

{

float per=(m1+m2+m3+m4+m5)/500\*100;

System.out.println("Percentage="+per);

}

public static void main(String args[])

{

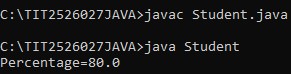
Student s= new Student();

s.calculate(80,80,80,80,80);

}

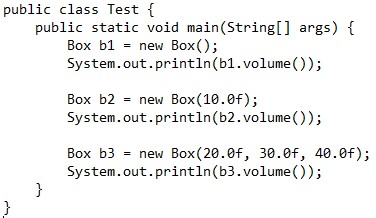
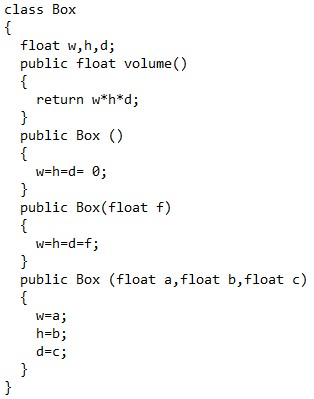
}

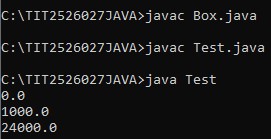
# OUTPUT



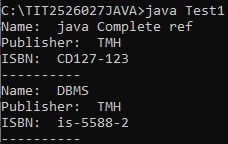
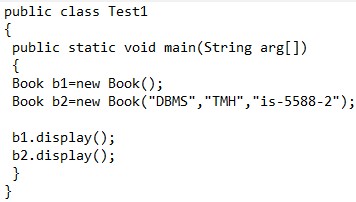
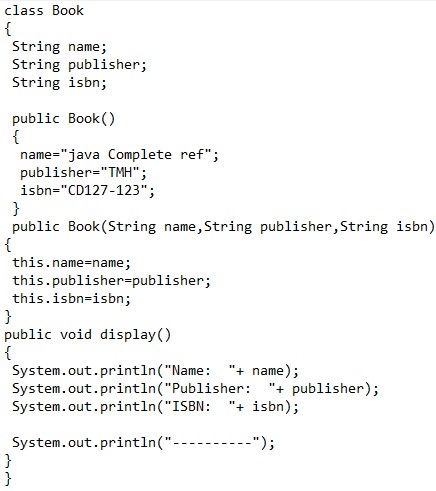
**Practical 3**

Q1)Write a java program to implement constructor overloading





Q2) Write a java program to demonstrate the use of this keyword ; both as an object and as a constructor

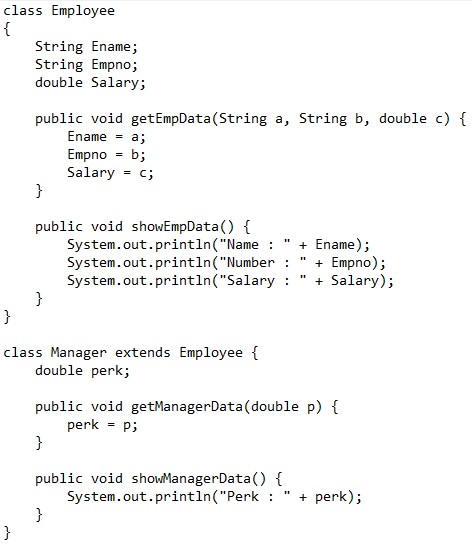


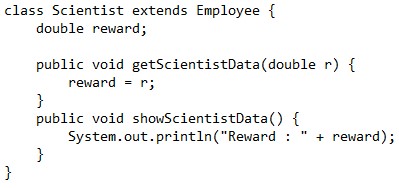
**Practical No 4: Inheritance**

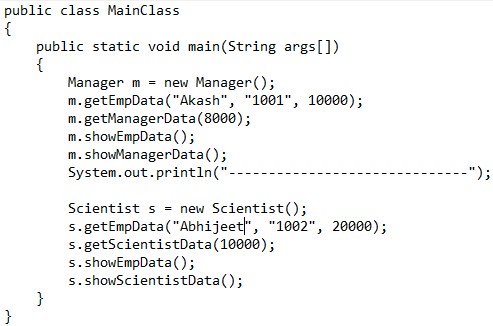
Q1) Write a java program for the following inheritance relationship Employee (super class) with 2 subclasses – Manager and Scientist Employee- variables , empno,ename,salary – getEmpData(),showEmpData() getEmpData() takes parameters to assign empno,ename,salary. showEmpData() displays the details

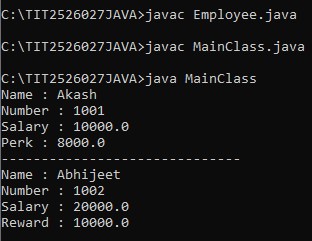
Manager – variable – perks – getMgrData(), showMgrData() getMgrData() takes parameters to assign perks showMgrData() displays the details

Scientist – variable – rewards – getSciData(), showSciData() getSciData() takes parameters to assign rewards.

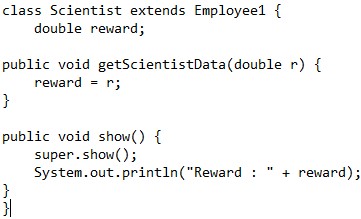
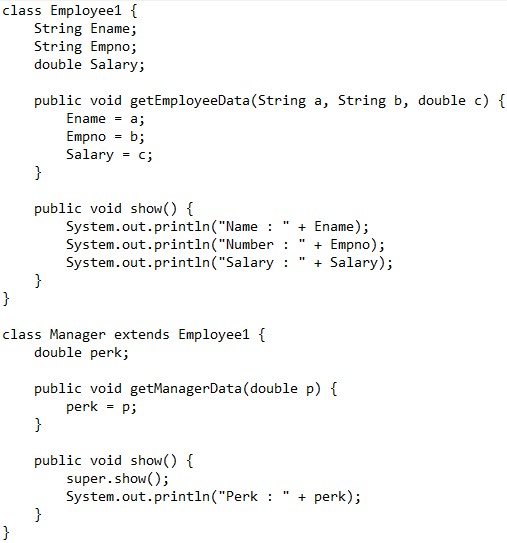


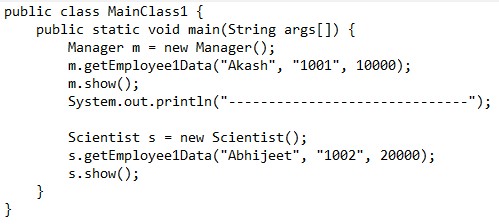


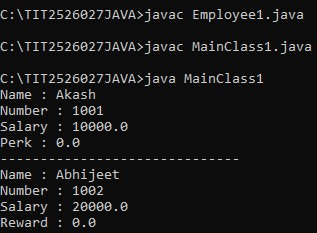




Q2) Modify Q1 to demonstrate super for method overriding. (Use show() for all classes)







**Practical No 5: Inheritance**

Q1) Modify Q2 in Practical 4 to Input values for all the classes using constructors.  class Employee

{

int empno; String ename; double salary;

public Employee(int empno,String ename,int salary)

{

this.empno=empno;

this.ename=ename; this.salary=salary;

}

void show()

{

System.out.println("Employee no"+empno);

System.out.println("Employee name"+ename);

System.out.println("Salary"+salary);

}

}

class Manager extends Employee

{

int perks;

public Manager(int empno,String ename,int salary, int perks)

{

super(empno,ename,salary); this.perks=perks;

}

void show()

{

super.show();

System.out.println("perks"+perks);

}

}

class Scientist extends Employee

{

int rewards;

public Scientist(int empno,String ename,int salary,int rewards)

{

super(empno,ename,salary);

this.rewards=rewards;

}

void show()

{

super.show();

System.out.println("Rewards:"+rewards);

}

}

class P4C

{

public static void main(String args[])

{

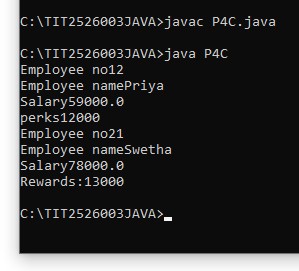
Manager ob=new Manager(12,"Priya",59000,12000);

ob.show();

Scientist s=new Scientist(21,"Swetha",78000,13000); s.show();

}

}



Q2) Write a java program to implement following inheritance relationship  The class PERSON has data member,name

The class STAFF is a subclass of Person and has data member; Id  The class TEMPORARY STAFF is a subclass of Staff and has data  members; no. of days worked, no.of hours worked

Input values for the above classes using constructors and display them  through a display function

class person

{

String name; person(String s)

{

name=s;

}

public void display()

{

System.out.println("name of person="+name);

}

}

class staff extends person

{

int id; staff(String name,int id)

{

super(name); this.id=id;

}

public void display()

{

super.display();

System.out.println("staff id is ="+id);

}

}

class temporarystaff extends staff

{

int days; int hoursworked;

temporarystaff(String sname,int id,int d,int h)

{

super(sname,id); days=d; hoursworked=h;

}

public void display()

{

System.out.println("no of days="+days);

System.out.println("no of hours worked="+hoursworked);

System.out.println("total salary="+days\*hoursworked\*500);

}

}

public class Test4

{

public ststic void main(String args[])

{

temporarystaff ts=new temporarystaff("abc",999,10,9);

ts.display();

}

}

public class Test4

{

public static void main(String args[])

{

temporarystaff ts=new temporarystaff("abc",999,10,9);

ts.display();

}

}

**PRACTICAL 6**

1. Write a java program to create the following classes:-

i) Create a class Figure with 2 variables dim1 and dim2 to store its dimensions and initialize the variables through its parameterized constructor. Include an abstract method area() ii) Create a class Rectangle which inherits Figure and implements the abstract method area() iii) Create a class Triangle which inherits Figure and implements the abstract method area() iv) Create a class which contains the main method and calls area for Rectangle and Triangle

abstract class Figure

{

double d1, d2;

public Figure(double a,double b)

{

d1 = a;

d2 = b;

}

abstract void area();

}

class Rectangle extends Figure

{

public Rectangle(double a,double b)

{

super(a,b);

}

void area()

{

System.out.println("The area of the rectangle is: "+(d1\*d2));

}

}

class Triangle extends Figure

{

public Triangle(double a,double b)

{

super(a,b);

}

void area()

{

System.out.println("The area of the triangle is: "+(0.5\*d1\*d2));

}

}

import java.util.Scanner;

class Demo

{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

System.out.println("Enter the length of the rectangle: "); double a = s.nextDouble();

System.out.println("Enter the breadth of the rectangle: ");

double b = s.nextDouble(); Rectangle r = new Rectangle(a,b); r.area();

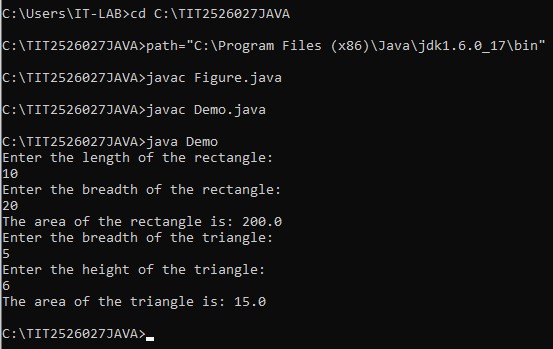
System.out.println("Enter the breadth of the triangle: "); double c = s.nextDouble();

System.out.println("Enter the height of the triangle: ");

double d = s.nextDouble(); Triangle t = new Triangle(c,d); t.area();

}

}



2. Rewrite the abstract class program using interfaces interface Figure1

{

public void area();

}

class Rectangle1 implements Figure1

{

double d1,d2;

public Rectangle1(double a, double b)

{

d1 = a; d2 = b;

}

public void area()

{

System.out.println("The area of the rectangle is: "+(d1\*d2));

}

}

class Triangle1 implements Figure1

{

double d1,d2;

public Triangle1(double a, double b)

{

d1 = a; d2 = b;

}

public void area()

{

System.out.println("The area of the triangle is: "+(0.5\*d1\*d2));

}

}

import java.util.Scanner;

class Demo1

{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

System.out.println("Enter the length of the rectangle: "); double a = s.nextDouble();

System.out.println("Enter the breadth of the rectangle: ");

double b = s.nextDouble(); Rectangle1 r = new Rectangle1(a,b); r.area();

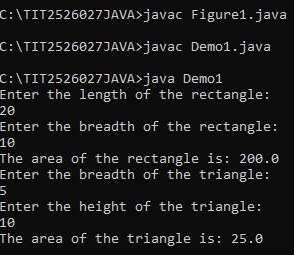
System.out.println("Enter the breadth of the triangle: "); double c = s.nextDouble();

System.out.println("Enter the height of the triangle: ");

double d = s.nextDouble(); Triangle1 t = new Triangle1(c,d); t.area();

}

}



Practical 7

1. Write a java program to demonstrate label import javax.swing.\*; class LabelDemo extends JFrame

{

public static void main(String[]args)

{

JFrame f=new JFrame("Lable Demo");

ImageIcon i=new ImageIcon("siesascs.jpg");

JLabel l=new JLabel("welcome to sies",i,JLabel.HORIZONTAL);

f.add(l);

f.setSize(600,500);

f.setVisible(true);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}



2. Write a java program to input the name ,class and college from the student using labels and text boxes and display “Hello<name>, you are admitted to <class> at <college> college” when the button is clicked

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

class ButtonDemo extends JFrame implements ActionListener

{

private JLabel l1,l2,l3; private JTextField txt1,txt2,txt3; private JButton b; private String name=" "; private String cl=" "; private String college=" ";

public ButtonDemo()

{

setLayout(new FlowLayout()); setSize(600,500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); l1=new JLabel("Enter your name:"); txt1=new JTextField(20); l2=new JLabel("Enter your class:"); txt2=new JTextField(20); l3=new JLabel("Enter your college:"); txt3=new JTextField(20); b=new JButton("SUBMIT"); b.addActionListener(this); add(l1); add(txt1);

add(l2); add(txt2); add(l3); add(txt3); add(b); setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b)

{

name=txt1.getText(); cl=txt2.getText(); college=txt3.getText();

JOptionPane.showMessageDialog(this,"Hello"+name+"You have taken admission for"+cl+"at"+college+"college");

System.exit(0);

}

} }

public class ButtonProg

{

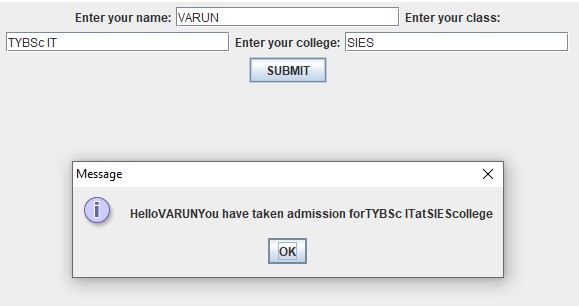
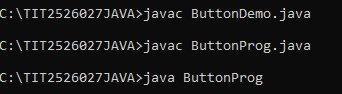
public static void main(String args[])

{

ButtonDemo ob=new ButtonDemo(); ob.setVisible(true);

}

}



**Practical 8**

# Q1)Java Program Using JcheckBox

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*;

class CheckBoxFrame extends JFrame implements ItemListener

{ private JTextField txt; private JCheckBox b; private JCheckBox i;

public CheckBoxFrame()

{

setLayout(new FlowLayout()); setSize(300,250);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

txt=new JTextField("Watch the font style change",20); txt.setFont(new Font("Serif",Font.PLAIN,14));

add(txt);

b=new JCheckBox("Bold"); i=new JCheckBox("Italic");

add(b); add(i);

b.addItemListener(this);

i.addItemListener(this); setVisible(true);

}

public void itemStateChanged(ItemEvent event)

{

Font font=null;

if(b.isSelected() && i.isSelected()) font=new Font("Serif",Font.BOLD+Font.ITALIC,14); else if(b.isSelected()) font=new Font("Serif",Font.BOLD,14); else if(i.isSelected()) font=new Font("Serif",Font.ITALIC,14); else

font=new Font("Serif",Font.PLAIN,14);

txt.setFont(font);

}

}

public class CheckBoxExample

{

public static void main(String args[])

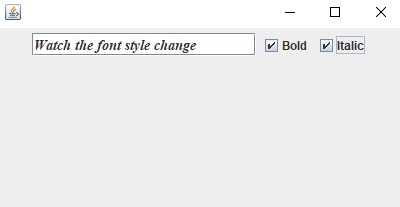
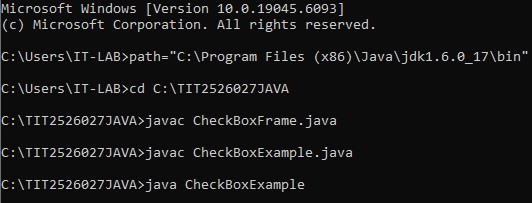
{

CheckBoxFrame c= new CheckBoxFrame();

c.setVisible(true);

}

}



# Q2)Java Program using RadioButton

import java.awt.event.\*;

import javax.swing.\*;

class RadioButtonExample extends JFrame implements ActionListener {

JRadioButton rb1, rb2;

JButton b;

public RadioButtonExample()

{

rb1 = new JRadioButton("FYBA"); rb1.setBounds(100, 50, 100, 30); rb2 = new JRadioButton("FYBscIT");

rb2.setBounds(100, 100, 100, 30);

ButtonGroup bg = new ButtonGroup();

bg.add(rb1); bg.add(rb2);

b = new JButton("Click");

b.setBounds(100, 150, 80, 30);

b.addActionListener(this);

add(rb1); add(rb2);

add(b);

setSize(300, 300);

setLayout(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (rb1.isSelected()) {

JOptionPane.showMessageDialog(this, "The credit course is Creative Writing");

}

if (rb2.isSelected()) {

JOptionPane.showMessageDialog(this, "The Credit Course is Cyber Security");

}

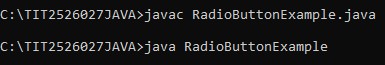
}

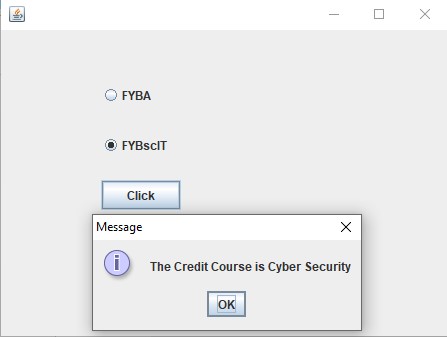
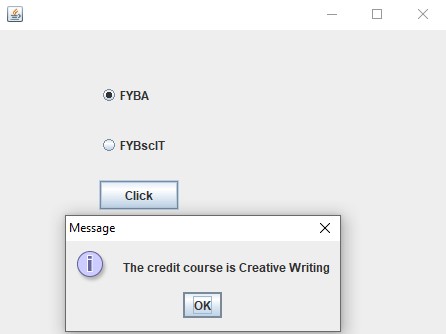
public static void main(String args[]) {

new RadioButtonExample();

}

}





Practical 9

Q1)Write a java program to implement Threads using the Thread class.

class A extends Thread

{

public void run()

{

for(int i=1;i<=5;i++)

{

System.out.println("Thread A with i"+i);

}

System.out.println("Exit from Thread A");

}

}

class B extends Thread

{

public void run()

{

for(int j=1;j<=5;j++)

{

System.out.println("Thread B with j"+j);

}

System.out.println("Exit from Thread B");

}

}

class C extends Thread

{

public void run()

{

for(int k=1;k<=5;k++)

{

System.out.println("Thread C with k"+k);

}

System.out.println("Exit from Thread C");

}

}

public class Prac8

{

public static void main(String args[])

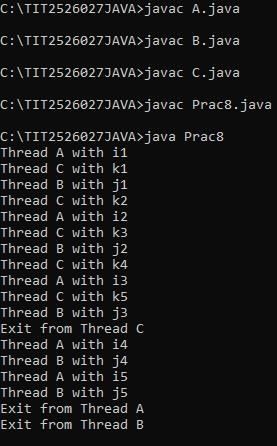
{

1. ob1=new A();
2. ob2=new B();
3. ob3=new C();

ob1.start(); ob2.start(); ob3.start();

}

}



Q2)Write a java program to implements threads using Runnable Interface

class A1 implements Runnable

{

public void run()

{

for(int i=1;i<=5;i++)

{

System.out.println("Thread A1 with i"+i);

}

System.out.println("Exit from Thread A1");

}

}

class B1 implements Runnable

{

public void run()

{

for(int j=1;j<=5;j++)

{

System.out.println("Thread B1 with j"+j);

}

System.out.println("Exit from Thread B1");

}

}

class C1 implements Runnable

{

public void run()

{

for(int k=1;k<=5;k++)

{

System.out.println("Thread C1 with k"+k);

}

System.out.println("Exit from Thread C1");

}

} public class Prac8B

{

public static void main(String args[])

{

1. ob1=new A();

Thread t1=new Thread(ob1);

1. ob2=new B();

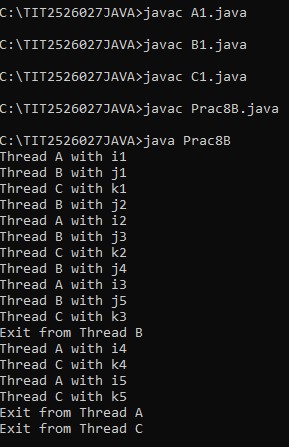
Thread t2=new Thread(ob2);

1. ob3=new C(); Thread t3=new Thread(ob3);

t1.start(); t2.start(); t3.start();

}

}



Q3)Write a java program to implements Threads using Runnable Interface(set priorities)

class A2 implements Runnable

{

public void run()

{

for(int i=1;i<=5;i++)

{

System.out.println("Thread A2 with i"+i);

}

System.out.println("Exit from Thread A2");

}

}

class B2 implements Runnable

{

public void run()

{

for(int j=1;j<=5;j++)

{

System.out.println("Thread B2 with j"+j);

}

System.out.println("Exit from Thread B2");

}

}

class C2 implements Runnable

{

public void run()

{

for(int k=1;k<=5;k++)

{

System.out.println("Thread C2 with k"+k);

}

System.out.println("Exit from Thread C2");

}

} public class Prac8C

{

public static void main(String args[])

{

1. a=new A();

Thread t1=new Thread(a);

1. b=new B();

Thread t2=new Thread(b);

1. c=new C(); Thread t3=new Thread(c); t3.setPriority(Thread.MAX\_PRIORITY); t2.setPriority(t3.getPriority()-1);

t1.setPriority(Thread.MIN\_PRIORITY);

t3.start(); t2.start(); t1.start();

}

}

