

B.M.S. College of Engineering
(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering



JAVA LAB REPORT

**SAATVIK S
1BM21CS178
SECTION:3C
BATCH:C4**

PROGRAM 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
public class QuadraticEquationExample1
{
    public static void main(String[] Strings)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the coefficients a,b,c: ");
        double a = s.nextDouble();
        double b = s.nextDouble();
        double c = s.nextDouble();
        double d= b * b - 4.0 * a * c;
        if (d> 0.0)
        {
            System.out.println("Real and distinct roots");
            double r1 = (-b + Math.pow(d, 0.5)) / (2.0 * a);
            double r2 = (-b - Math.pow(d, 0.5)) / (2.0 * a);
            System.out.println("The roots are " + r1 + " and " + r2);
        }
        else if (d == 0.0)
        {
            System.out.println("Real and equal roots");
            double r1 = -b / (2.0 * a);
            System.out.println("r1=r2= " + r1);
        }
        else
        {
            System.out.println("Imaginary roots.");
        }
    }
}
```

OUTPUT:

The image shows a Windows desktop environment. In the top-left corner, there is a Notepad window titled "Quadratic - Notepad" containing Java code for solving quadratic equations. In the bottom-right corner, there is a terminal window titled "C:\Windows\System32\cmd.exe" showing the execution of the Java program and its output.

```
Quadratic - Notepad
File Edit Format View Help
import java.util.Scanner;
import java.lang.Math;
class Quadratic
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int a,b,c;
        double d,r1,r2;
        System.out.println("Enter coeffiencts a,b,c:");
        a=s.nextInt();
        b=s.nextInt();
        c=s.nextInt();
        if(a==0)
            System.out.println("Invalid Equation");
        else
            d=(b*b-4*a*c);
        if(d==0)
            System.out.println("Real and equal roots\n");
        System.out.println("r1=r2=" +(-b/2*a));
        else if(d>0)
            r1=(-b+Math.sqrt(d)/2*a);
            r2=(-b-Math.sqrt(d)/2*a);
            System.out.println("Real and distinct roots\n");
            System.out.println("r1=" +r1+"r2=" +r2);
        else if(d<0)
            System.out.println("Imaginary roots\n");
            System.out.println("r1=" +(-b/2*a)+"+" +i+(Math.sqrt(-d)/2*a));
            System.out.println("r2=" +(-b/2*a)+"-" +i+(Math.sqrt(-d)/2*a));
    }
}
```

```
C:\Users\bmsce\Desktop>javac Quadratic.java
C:\Users\bmsce\Desktop>java Quadratic
Enter coeffiencts a,b,c:
0 1 2
Invalid Equation

C:\Users\bmsce\Desktop>java Quadratic
Enter coeffiencts a,b,c:
1 6 9
Real and equal roots
r1=r2=-3

C:\Users\bmsce\Desktop>java Quadratic
Enter coeffiencts a,b,c:
1 6 5
Real and distinct roots
r1=-4.0r2=-8.0

C:\Users\bmsce\Desktop>java Quadratic
Enter coeffiencts a,b,c:
1 6 10
Imaginary roots
r1=-3+i1.0
r2=-3-i1.0

C:\Users\bmsce\Desktop>
```

Activate Windows
Go to Settings to activate Windows.

Type here to search

Ln 32, Col 62 100% Windows (CRLF) UTF-8 03:49 PM 18-11-2022

OBSERVATION

Java Lab

Prog 1

Prog to find roots of quadratic equation

```
import java.util.Scanner;  
import java.lang.Math;
```

```
class Quadratic
```

```
{
```

```
    public static void main (String args [])
```

```
{
```

```
    Scanner s = new Scanner (System.in);
```

```
    int a,b,c;
```

```
    double d,r1,r2;
```

```
    System.out.println ("Enter coefficients a,b,c:");
```

```
    a=s.nextInt();
```

```
    b=s.nextInt();
```

```
    c=s.nextInt();
```

```
{
```

```
    if (a==0)
```

```
        System.out.println ("Invalid Equation");
```

```
    else
```

```
{
```

```
    d=b*b - 4*a*c;
```

```
    if (d==0)
```

```
{
```

```
        System.out.println ("Real and equal roots");
```

```
        System.out.println ("r1=r2=" + (-b/2*a));
```

```
}
```

```
    else if (d>0)
```

```
{
```

```
        r1= (-b + Math.sqrt(d)/2*a);
```

```
        r2= (-b - Math.sqrt(d)/2*a);
```

System.out.println ("Real and distinct roots \n");

System.out.println ("r₁ = " + r1 "r₂ = " + r2);

}

else if (d < 0)

{

System.out.println ("Imaginary roots \n");

System.out.println ("r₁ = " +

Output

1. Enter the coefficients a, b, c;

1 6 9

Real and equal roots

r₁ = r₂ = -3

2. Enter the coefficients a, b, c;

1 6 5

Real and distinct roots

r₁ = -4.0 r₂ = -8.0

3. Enter the coefficients a, b, c;

1 6 10

Imaginary roots

r₁ = -3 + i1.0

r₂ = -3 - i1.0

Ques
Ans
18/11/22

PROGRAM 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Studentx{
    Scanner s=new Scanner(System.in);
    private int usn;
    private String name;
    int n;
    int[] marks=new int[10];
    int[] credits=new int[10];
    void getMarks(){
        System.out.println("Enter no of subjects:");
        this.n=s.nextInt();
        System.out.println("Enter marks of subjects and credits consecutively:");
        for(int i=0;i<n;i++){
            this.marks[i]=s.nextInt();
            this.credits[i]=s.nextInt();
        }
    }
    int[] getGrade(){
        int[] grade=new int[10];
        for(int i=0;i<n;i++){
            if(marks[i]>=90){
                grade[i]=10;
            }else if(marks[i]>=80){
                grade[i]=9;
            }else if(marks[i]>=70){
                grade[i]=8;
            }else if(marks[i]>=60){
                grade[i]=7;
            }else if(marks[i]>=50){
                grade[i]=6;
            }else if(marks[i]>=40){
                grade[i]=5;
            }else if(marks[i]>=30){
                grade[i]=4;
            }else{
                grade[i]=0;
            }
        }
        return grade;
    }
}
```

```

    }
    double calc(){
        int sum1=0,sum2=0;
        for(int i=0;i<n;i++){
            int[] grade;
            grade=getGrade();
            sum1+=grade[i]*this.credits[i];
            sum2+=this.credits[i];
        }
        return (double)sum1/sum2;
    }
    void display(){
        System.out.println("Marks\tCredits\n");
        for(int i=0;i<n;i++){
            System.out.println(this.marks[i]+"\t"+this.credits[i]+"\n");
        }
        System.out.println("SGPA="+calc());
    }
}
class mainx{
    public static void main(String args[]){
        Studentx s=new Studentx();
        s.getMarks();
        s.calc();
        s.display();
    }
}

```

OUTPUT:

```

Enter no of subjects:
4
Enter marks of subjects and credits consecutively:
80 3
90 4
85 3
80 2
Marks  Credits

80      3

90      4

85      3

80      2

SGPA=9.33333333333334

Process finished with exit code 0

```

Program - 2

```
import java.util.Scanner;  
  
class Student {  
    Scanner s = new Scanner(System.in);  
    private int num;  
    private String name;  
    int n;  
    int[] marks = new int[10];  
    int[] credits = new int[]{};  
    void getMarks() {  
        System.out.println("Enter no. of subjects:");  
        this.n = s.nextInt();  
        System.out.println("Enter marks & credits of corresponding  
                           subject");  
        for (int i=0; i<n; i++) {  
            this.marks[i] = s.nextInt();  
            this.credits[i] = s.nextInt();  
        }  
    }  
  
    int[] getGrade() {  
        int[] grade = new int[10];  
        for (int i=0; i<n; i++) {  
            if (marks[i] >= 90) {  
                grade[i] = 10;  
            } else if (marks[i] >= 80) {  
                grade[i] = 9;  
            } else if (marks[i] >= 70) {  
                grade[i] = 8;  
            } else if (marks[i] >= 60) {  
                grade[i] = 7;  
            }  
        }  
        return grade;  
    }  
}
```

```
else if (marks[i] >= 50) {  
    grade[i] = 6;  
} else if (marks[i] >= 40) {  
    grade[i] = 5;  
} else if (marks[i] >= 40) {  
    grade[i] = 4;  
}  
else {  
    grade[i] = 0;  
}  
}  
  
return grade;
```

```
double calc() {  
    int sum1 = 0, sum2 = 0;  
    for (int i = 0; i < n; i++) {  
        int [] grade;  
        grade = getGrade();  
        sum1 += grade[i] * this.credits[i];  
        sum2 += this.credits[i];  
    }  
    return (double) sum1 / sum2;  
}
```

```
void display() {  
    System.out.println ("Marks |t Credits |n");  
    for (int i = 0; i < n; i++) {  
        System.out.println (this.marks[i] |t|t + this.credits[i] + " |n");  
    }  
    System.out.println ("SGPN = " + calc());  
}
```

```
public static void main (String args []) {  
    Student s = new Student ();  
    s.getMarks ();  
    s.calc ();  
    s.display;  
}
```

OUTPUT :

Enter no of subjects :

4

Enter marks of Subjects and credits consecutively.

80 3

90 4

85 3

80 2

Marks	Credits
80	3
90	4
85	3
80	2

SGPA = 9.3333

PROGRAM 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.io.*;
import java.util.*;
class Book{
    String Title,Author;
    int pages;
    double price;

    Book()
    {
        Title="xyz";
        Author="abc";
        price=0.0;
        pages=0;
    }

    void Set_Title(String t)
    {
        Title=t;
    }

    void Set_Author(String a)
    {
        Author=a;
    }

    void Set_price(double p)
    {
        price=p;
    }

    void Set_pages(int x)
    {
        pages=x;
    }

    public String toString()
    {
```

```

        return Title+"\t"+Author+"\t"+price+"\t"+pages+"\n";
    }
}

class Details
{
    public static void main(String args[])
    {
        String t,a;
        double p;
        int x,n,i;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the no of books:");
        n=s.nextInt();
        Book b[]=new Book[n];
        for(i=0;i<n;i++)
        {
            System.out.println("Enter the title of the book:");
            t=s.next();
            System.out.println("Enter the author of the book:");
            a=s.next();
            System.out.println("Enter the price of the book:");
            p=s.nextDouble();
            System.out.println("Enter the no of pages of the book:");
            x=s.nextInt();
            b[i]= new Book();
            b[i].Set_Title(t);
            b[i].Set_Author(a);
            b[i].Set_price(p);
            b[i].Set_pages(x);
        }
        System.out.println("Title\tAuthor\tPrice\tPages\n");
        for(i=0;i<n;i++)
        {
            System.out.println(b[i]);
        }
    }
}

```

OUTPUT:

The image shows a Windows desktop environment. In the foreground, there is a Notepad window titled "Details.java - Notepad" containing Java code. The code defines a class Details with a main method that reads input for multiple books and prints them to the console. Below the Notepad is a Command Prompt window titled "C:\Windows\System32\cmd.exe" showing the execution of the Java program and its output.

```
File Edit Format View Help
}
class Details
{
    public static void main(String args[])
    {
        String t,a;
        double p;
        int x,n,i;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the no of books:");
        n=s.nextInt();
        Book b[] = new Book[n];
        for(i=0;i<n;i++)
        {
            System.out.println("Enter the title of the book:");
            t=s.next();
            System.out.println("Enter the author of the book:");
            a=s.next();
            System.out.println("Enter the price of the book:");
            p=s.nextDouble();
            System.out.println("Enter the no of pages of the book:");
            x=s.nextInt();
            b[i]= new Book();
            b[i].Set_Title(t);
            b[i].Set_Author(a);
            b[i].Set_Price(p);
            b[i].Set_Pages(x);
        }
        System.out.println("Title\tAuthor\tPrice\tPages\n");
        for(i=0;i<n;i++)
        {
            System.out.println(b[i]);
        }
    }
}

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Student\Desktop\GS178>javac Details.java
C:\Users\Student\Desktop\GS178>java Details
Enter the no of books:
2
Enter the title of the book:
abc
Enter the author of the book:
pqr
Enter the price of the book:
120
Enter the no of pages of the book:
300
Enter the title of the book:
lmn
Enter the author of the book:
def
Enter the price of the book:
200
Enter the no of pages of the book:
450
Title      Author     Price    Pages
abc        pqr       120.0    300
lmn        def       200.0    450
```

```
import java.io.*;  
import java.util.*;
```

```
class Book {
```

```
    String Title, Author;
```

```
    int pages;
```

```
    double price;
```

```
    Book ()
```

```
{
```

```
    Title = "xyz";  
    Author = "abc";  
    price = 0.0;  
    pages = 0;
```

```
}
```

```
void Set-Title (String t)
```

```
{
```

```
    Title = t;
```

```
}
```

```
void Set-Author (String a)
```

```
{
```

```
    Author = a;
```

```
}
```

```
void Set-price (double p)
```

```
{
```

```
    price = p;
```

```
}
```

```
void Set-pages (int x)
```

```
{
```

```
    pages = x;
```

```
}
```

```
public String toString ()  
{  
    return Title + "\t" + Author + "\t" + Price + "\t" + Pages + "\n";  
}  
}
```

class Details

```
{
```

```
public static void main (String args [])
```

```
{
```

```
String t, a;
```

```
double p;
```

```
int x, n, i;
```

```
Scanner s = new Scanner (System.in);
```

```
System.out.println ("Enter no. of books : ");
```

```
n = s.nextInt ();
```

```
Book b [] = new Book [n];
```

```
for (i=0; i<n; i++)
```

```
{
```

```
System.out.println ("Enter the title of the book : ");
```

```
t = s.next();
```

```
System.out.println ("Enter the author of the book : ");
```

```
a = s.next();
```

```
System.out.println ("Enter the price of the book : ");
```

```
p = s.nextDouble();
```

```
System.out.println ("Enter no. of pages : ");
```

```
x = s.nextInt();
```

```
b[i] = new Book();
```

```
b[i].Set>Title (t);
```

```
b[i].SetAuthor (a);
```

```
b[i].SetPrice (p);
```

```
b[i].SetPages (x);
```

```
}
```

```
System.out.println("Title |t Author |t Price |t Pages |n");
```

```
for (i=0; i<n; i++)
```

```
{
```

```
    System.out.println(b[i]);
```

```
}
```

```
}
```

O/P

Enter the no. of books :

2

Enter the author of book :

pqr

Enter the title of book :

abc

Enter the no. of pages :

120

Enter the price of book :

300

Enter the title of book :

lmn

Enter the author of book :

def

Enter the price of book :

200

Enter the no. of pages of book :

450

Title	Author	Price	Pages
abc	pqr	120.0	300
lmn	def	200.0	450

Output : 21/12/22

Author Name : Suman Singh

Date : 21/12/22

PROGRAM 4

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

```
import java.util.Scanner;
abstract class Shape{
    int l;
    int b;
    public abstract void printArea();
}

class Rectangle extends Shape{
    public void printArea(){
        System.out.println("enter values");
        Scanner sc = new Scanner(System.in);
        int l = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(l*b);
    }
}

class Triangle extends Shape{
    public void printArea(){
        System.out.println("enter values");
        Scanner sc = new Scanner(System.in);
        int l = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(0.5*l*b);
    }
}

class Circle extends Shape{
    public void printArea(){
        System.out.println("enter values");
        Scanner sc = new Scanner(System.in);
        int l = sc.nextInt();
        System.out.println((3.14)*l*l);
    }
}
```

```
public class Area{  
  
    public static void main(String args[]){  
  
        Rectangle r = new Rectangle();  
        r.printArea();  
        Triangle t = new Triangle();  
        t.printArea();  
        Circle c = new Circle();  
        c.printArea();  
    }  
}
```

OUTPUT:

```
enter values  
10 20  
200  
enter values  
12 10  
60.0  
enter values  
5  
78.5  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

```
import java.util.*;  
abstract class shape {  
    int a, b;  
  
    public Shape (int a, int b) {  
        this.a = a;  
        this.b = b;  
    }  
    abstract void printarea();  
}  
  
class Circle extends Shape {  
    Circle (int a, int b) {  
        Super (a, b);  
    }  
    void printarea () {  
        System.out.println ("Area of circle is " + (3.14*a*a));  
    }  
}  
  
class Rectangle extends Shape {  
    public Rectangle (int a, int b) {  
        super(a,b);  
    }  
    void printarea () {  
        System.out.println ("Area of rectangle is " + (a*b));  
    }  
}  
  
class Triangle extends Shape {  
    public Triangle (int a, int b) {  
        super (a,b);  
    }  
}
```

```
void printarea() {
```

```
    System.out.println("Area of Triangle is : " + (0.5 * a * b));
```

```
}
```

```
class main {
```

```
    public static void main (String args[]) {
```

```
        Scanner s = new Scanner (System.in);
```

```
        System.out.println ("Enter dim 1 : ");
```

```
        int x = s.nextInt();
```

```
        System.out.println ("Enter dim 2 : ");
```

```
        int y = s.nextInt();
```

```
        Shape p;
```

```
        p = new Circle (x,y);
```

```
        p.printarea();
```

```
        p = new Rectangle (x,y);
```

```
        p.printarea();
```

```
        p = new Triangle (x,y);
```

```
        p.printarea();
```

```
}
```

O/P

Enter dim 1 : 10

Enter dim 2 : 12

Area of circle : 5024

Area of Rectangle : 720

Area of triangle : 270.0

PROGRAM 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the 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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

```
import java.util.Scanner;
import java.lang.Math;
class Account
{
    String name, acc_type;
    int acc_no;
    double bal,dep;
    Scanner scan= new Scanner(System.in);
    void setd()
    {
        System.out.println("Enter your Name:");
        name=scan.next();
        System.out.println("Enter your Account Number:");
        acc_no=scan.nextInt();
        System.out.println("Enter your Account type: (Savings/Current)");
        acc_type=scan.next();
        System.out.println("Enter the Bank Balance:");
        bal=scan.nextInt();
    }
    void disp()
    {
        System.out.println("Name: "+name);
        System.out.println("Account Number: "+acc_no);
        System.out.println("Account Type: "+acc_type);
        System.out.println("Current balance is: "+bal);
    }
    void deposit()
    {
```

```
System.out.println("Enter the amount to be deposited:");
dep=scan.nextInt();
bal+=dep;
System.out.println("BALANCE AMOUNT: "+bal);
}
/*boolean acc(String acc_type)
{
if(acc_type.equals("Savings"))
return true;
else if(acc_type=="Current")
return false;
else
return true;
}*/
}
class Cur_acct extends Account
{
int penal()
{
double min, pen;
System.out.println("Enter Minimum balance & penalty amount if not followed:");
min=5000; pen=min*0.05;
if(bal<min)
{
bal-=pen;
System.out.println("Penalty imposed for having insufficient balance"); return 0;
}
else
{
System.out.println("No penalty");
return 1;
}
}
void withdrawal()
{
double amt;
System.out.println("Enter amount to be withdrawn:");
amt=scan.nextInt();
int a= penal();
if(a==1)
{
if(bal>=amt)
{ bal=bal-amt;
System.out.println("Account Balance after withdrawal is:" +bal);}
}
else
```

```

        System.out.println("The amount can't be withdrawn");
    }
}
class Sav_acct extends Account
{
void calc_interest()
{

double CI = bal*0.06;
System.out.println("Compounding interest: "+ CI);
}
void withdrawal()
{
double amt;
System.out.println("Enter amount to be withdrawn:");
amt=scan.nextInt();
if(bal>=amt)
{
    bal=bal-amt;
}
System.out.println("Account Balance after withdrawal is:" +bal);
else
    System.out.println("The amount can't be withdrawn");
}
}
class Bank
{
public static void main(String arg[])
{
Scanner ss=new Scanner(System.in);
Account b1=new Account();
b1.setd();
if(b1.acc_type.equals("Savings"))
{
Sav_acct s1=new Sav_acct();
//s1=b1;
s1.name=b1.name; s1.acc_no=b1.acc_no; s1.acc_type=b1.acc_type; s1.bal=b1.bal;
while(true)
{
System.out.println("Enter your choice:\n1.Deposit\n2.Calculate
interest\n3.Withdraw\n4.Display\n5.Exit");
int choice=ss.nextInt();
switch(choice)
{
case 1: s1.deposit(); break;
case 2: s1.calc_interest(); break;
}
}
}
}

```

```
case 3: s1.withdrawal(); break;
case 4: s1.disp(); break;
case 5: System.exit(0);
default: System.out.println("Invalid input");
}
}
}
else if(b1.acc_type.equals("Current"))
{
Cur_acct c1=new Cur_acct();
c1.name=b1.name; c1.acc_no=b1.acc_no; c1.acc_type=b1.acc_type; c1.bal=b1.bal;
while(true)
{
System.out.println("Enter your choice:\n1.Deposit\n2.Penalty
Check\n3.Withdraw\n4.Display\n5.Exit");
int choice=ss.nextInt();
switch(choice)
{
case 1: c1.deposit(); break;
case 2: c1.penal(); break;
case 3: c1.withdrawal(); break;
case 4: c1.disp(); break;
case 5: System.exit(0);
default: System.out.println("Invalid input");
}
}
}
}
else
System.out.println("Invalid Account type");
}
}
```

OUTPUT:

The screenshot shows a Windows desktop environment. In the foreground, there is a Command Prompt window titled 'C:\Windows\System32\cmd.exe - java Bank'. The window displays a Java application for managing a bank account. The application asks for the account type ('Savings' or 'Current'), which is set to 'Savings'. It then asks for the initial bank balance, which is entered as '10000'. The application presents a menu with choices: 1. Deposit, 2. Calculate interest, 3. Withdraw, 4. Display, and 5. Exit. The user selects choice 3 (Withdraw) and attempts to withdraw '15000', but the application responds that this amount can't be withdrawn. The balance remains at '10000'. The user then selects choice 4 (Display), which shows the current balance as '10000.0'. The user then exits the application.

```
C:\Windows\System32\cmd.exe - java Bank
Enter your Account type: (Savings/Current)
Savings
Enter the Bank Balance:
10000
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
4. Display
5. Exit
3
Enter amount to be withdrawn:
15000
The amount can't be withdrawn
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
4. Display
5. Exit
4
Enter the amount to be deposited:
20000
BALANCE AMOUNT: 12000.0
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
4. Display
5. Exit
3
Enter amount to be withdrawn:
15000
The amount can't be withdrawn
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
4. Display
5. Exit
2
Compounding interest: 720.0
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
4. Display
5. Exit
4
Name: abc
Account Number: 123
Account Type: Savings
Current balance is: 12000.0
Enter your choice:
1. Deposit
2. Calculate interest
3. Withdraw
```

In the background, there is a Notepad window titled 'a - Notepad' which is currently empty.

```
import java.util.*;
import java.lang.Math;

class Account {
    String name, acc_type;
    int acc_no;
    double bal, amount;
    Scanner s = new Scanner(System.in);

    void info() {
        System.out.println("Enter name:");
        name = s.nextLine();
        System.out.println("Enter account no:");
        acc_no = s.nextInt();
        System.out.println("Enter account type:");
        acc_type = s.nextLine();
        System.out.println("Enter bank balance:");
        bal = s.nextInt();
    }

    void disp() {
        System.out.println("Name: " + name);
        System.out.println("Account Number: " + acc_no);
        System.out.println("Account type: " + acc_type);
        System.out.println("Current balance: " + bal);
    }

    void deposit() {
        System.out.println("Enter amount to be deposited:");
        dep = s.nextInt();
        bal += dep;
        System.out.println("Balance amount: " + bal);
    }
}
```

Class current extends Account

{

int penal()

{

double min, pen;

System.out.println("Enter min bal & penalty amount:");

min = 5000;

pen = min * 0.05;

if (bal < min)

{

bal = pen;

System.out.println("Penalty imposed.");

return 0;

}

else

{ System.out.println("No penalty");

return 1;

void withdrawl()

{

double amt;

System.out.println("Enter amount to be withdrawn:");

amt = sc.nextInt();

int a = penal();

if (a > 1)

{

if (bal >= amt)

{ bal -= amt;

}

}

```
class Savings extends Account
{
    void calcInterest()
    {
        double CI = bal * 0.06;
        System.out.println ("Compounding interest: " + CI);
    }

    void withdrawal ()
    {
        double amt;
        System.out.println ("Enter amount to be withdrawn:");
        amt = sc s.nextInt();
        if (bal >= amt)
        {
            bal = bal - amt;
            System.out.println ("Account balance after withdrawal : " + bal);
        }
        else
            System.out.println ("The amount can't be withdrawn");
    }
}
```

```
class Bank
```

```
{
```

```
public static void main (String args[])
{
```

```
Scanner sc = new Scanner (System.in);
```

```
Account b1 = new Account();
```

```
b1.setd();
```

```
if (b1.acc-type.equals ("Savings"))
```

```
{
```

```
Savings s1 = new Savings();
```

```
s1.name = b1.name;
```

```
s1.acc-no = b1.acc-no;
```

```
s1.acc-type = b1.acc-type;
```

```
s1.bal = b1.bal;
```

white(true)

(1) define

{

System.out.println ("Enter your choice : 1 n 2. Deposit In 3. calculate
interest In 4. Withdraw In 5. Display In
S. Exit ");

int c = s.nextInt();

Switch (c)

{

case 1: s1.deposit();

break;

case 2: s1.calc_interest();

break;

case 3: s1.withdraw();

break;

case 4: s1.disp();

break;

case 5: System.exit(0);

default: System.out.println ("Invalid input");

}

{

else if (bl.acc-type.equals ("Current"))

{

current c1 = new Current();

c1.name = bl.name;

c1.acc-no = bl.acc-no;

s1.acc-type = bl.acc-type;

s1.bal = bl.bal

white(true)

{

System.out.println ("Enter your choice : ");

int c = s.nextInt();

TUTORIAL

switch (c)

{

case 1 : cl. deposit();

break;

case 2 : cl. penal();

break;

case 3 : cl. withdrawal();

break;

case 4 : cl. disp();

break;

case 5 : system.exit(0);

default : System.out.println ("Invalid Input");

}

}

use

System.out.println ("Invalid account type");

}

OUTPUT

Enter your Account type : (Savings / current)

Savings

Enter the bank balance :

10000

Enter your choice :

1. Deposit

2. Calculate interest

3. Withdraw

4. Display

5. Exit.

1

Enter amount to be deposited :

2000

Balance amount : 12000.0

Enter your choice :

3

Enter amount to be withdrawn :

15000

The amount can't be withdrawn

Enter your choice :

2

Compounding interest : 720.0

Enter your choice

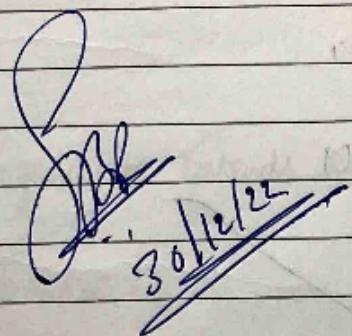
4

Name : abc

Account Number : 123

Account type : Savings

Current balance : 12000.0


30/12/22

PROGRAM 6

Anchor College offers both graduate and postgraduate programs. The college stores the names of the students, their test scores and the final result for each student. Each student has to take 4 tests in total. You need to create an application for the college by implementing the classes based on the class diagram and description given below.

```
import java.util.Scanner;
import java.util.Arrays;
interface std
{
    public String getName();
    public void setTestScore(int n,int marks);
    public int[] getTestScore();
    public void setTestResult();
    public int getTestResult();
    public void display();
}
abstract class Student implements std
{
    String name;
    int[] test = new int[4];
    int sum;

    abstract public void generateResult();

    Student()
    {}

    Student(String name)
    {
        this.name = name;
    }
    public String getName()
    {
        return this.name;
    }
    public void setTestScore(int n,int marks)
    {
        test[n-1] = marks;
    }
    public int[] getTestScore()
    {
        return test;
```

```
    }
    public void setTestResult()
    {
        for(int i=0;i<4;i++)
        {
            sum=sum+test[i];
        }
        sum/=4;
    }
    public int getTestResult()
    {
        return sum;
    }
    public void display()
    {
        System.out.println("\nStudent Name : "+getName());
        System.out.println("Student Marks : ");
        for(int i=0;i<4;i++)
            System.out.print(" "+test[i]);
        System.out.println("\nResult : ");
        generateResult();
    }
}
class UnderGraduate extends Student
{
    UnderGraduate()
    {}
    UnderGraduate(String name)
    {
        this.name = name;
    }
    public void generateResult()
    {
        if(getTestResult()>=60)
            System.out.print("Pass");
        else
            System.out.print("Fail");
    }
}
class Graduate extends Student
{
    Graduate()
    {}
    Graduate(String name)
```

```

{
    this.name = name;
}
public void generateResult()
{
if(getTestResult()>=70)

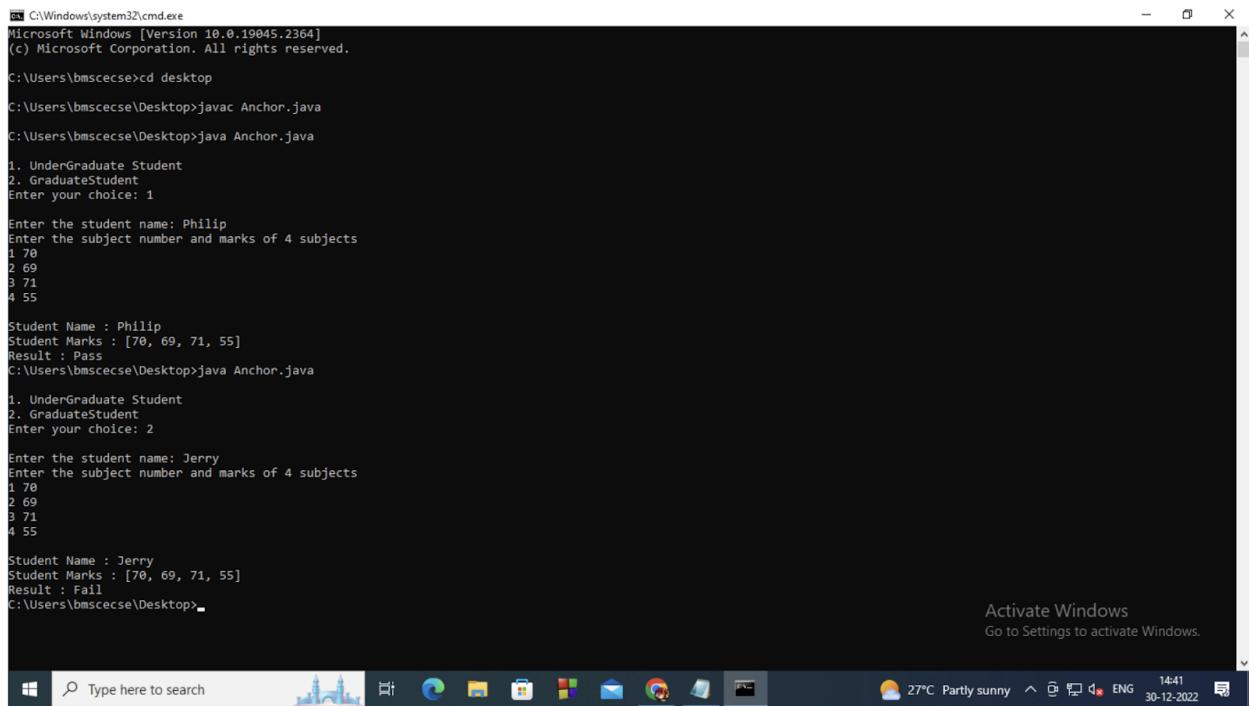
System.out.print("Pass");
else
System.out.print("Fail");
}
}

public class Stud
{
public static void main(String[] args)
{
Scanner s = new Scanner(System.in);
int c;
System.out.println("1. UnderGraduate Student\n2. Graduate Student\nEnter your choice:");
c = s.nextInt();
switch(c)
{
case 1: {
System.out.print("\nEnter the student name: ");
UnderGraduate u = new UnderGraduate(s.next());
System.out.println("Enter the subject number and marks of 4 subjects");
for(int i=0;i<4;i++)
{
u.setTestScore(s.nextInt(),s.nextInt());
}
u.setTestResult();
u.display();
}
break;
}
case 2: {
System.out.print("\nEnter the student name: ");
Graduate g = new Graduate(s.next());
System.out.println("Enter the subject number and marks of 4 subjects");
for(int i=0;i<4;i++)
{
g.setTestScore(s.nextInt(),s.nextInt());
}
g.setTestResult();
}
}
}

```

```
        g.display();
    }
    break;
default: System.out.println("Invalid Choice!");
}
}
}
```

OUTPUT:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsccecse>cd desktop
C:\Users\bmsccecse\Desktop>javac Anchor.java
C:\Users\bmsccecse\Desktop>java Anchor.java

1. UnderGraduate Student
2. GraduateStudent
Enter your choice: 1

Enter the student name: Philip
Enter the subject number and marks of 4 subjects
1 70
2 69
3 71
4 55

Student Name : Philip
Student Marks : [70, 69, 71, 55]
Result : Pass
C:\Users\bmsccecse\Desktop>java Anchor.java

1. UnderGraduate Student
2. GraduateStudent
Enter your choice: 2

Enter the student name: Jerry
Enter the subject number and marks of 4 subjects
1 70
2 69
3 71
4 55

Student Name : Jerry
Student Marks : [70, 69, 71, 55]
Result : Fail
C:\Users\bmsccecse\Desktop>
```

Activate Windows
Go to Settings to activate Windows.

```
import java.util.Scanner;  
import java.util.Arrays;  
interface Std  
{  
    public String getName();  
    public void setTestScore(int n, int marks);  
    public int[] getTestScore();  
    public void setTestResult();  
    public int getTestResult();  
    public void display();  
}
```

abstract class Student implements Std

```
{
```

```
String name;  
int[] test = new int[4];  
int sum;
```

```
abstract public void generateResult();  
Student()
```

```
{}
```

```
Student (String name)
```

```
{
```

```
this.name = name;
```

```
}
```

```
public String getName()
```

```
{
```

```
return this.name;
```

```
}
```

```
public void setTestScore (int n, int marks)
```

```
{
```

test[n-1] = marks;

```
}
```

```
public int[] getTestScore()
{
    return test;
}

public void setTestResult()
{
    for(int i=0; i<4; i++)
    {
        sum = sum + test[i];
    }
    sum /= 4;
}

public int getTestResult()
{
    result sum;
}

public void display()
{
    System.out.println("Student Name : " + getName());
    System.out.println("Student marks : ");
    for(int i=0; i<4; i++)
        System.out.print(" " + test[i]);
    System.out.println("Result : ");
    generateResult();
}
```

Class Undergraduate extends Student

```
Undergraduate()
{
}
```

```
Undergraduate(String name)
{
    this.name = name;
}
```

this.name = name;

```
public void generateResult ()
```

```
{
```

```
if (getTestResult () >= 60)
```

```
System.out.print ("Pass");
```

```
else
```

```
System.out.print ("Fail");
```

```
}
```

```
}
```

```
class Graduate extends Student
```

```
{
```

```
Graduate ()
```

```
{}
```

```
Graduate (String name)
```

```
{
```

```
this.name = name;
```

```
}
```

```
public void generateResult ()
```

```
{ if (getTestResult () >= 70)
```

```
System.out.print ("Pass");
```

```
else
```

```
System.out.print ("Fail");
```

```
}
```

```
public class Stud
```

```
{
```

```
public static void main (String [] args)
```

```
{
```

```
Scanner S = new Scanner (System.in)
```

```
int C;
```

```
System.out.println ("1. Undergraduate Student In
```

```
2. Graduate Student In Enter your choice:
```

```
switch(c)
```

```
{
```

```
case 1 : {
```

```
    System.out.print("Enter the student name : ");
```

```
    Undergraduate u = new Undergraduate(s.nextInt());
```

```
    System.out.print("Enter the subject number and marks of 4 subjects");
```

```
    for (i=0; i<4; i++)
```

```
{
```

```
        u.TestScore(s.nextInt(), s.nextInt());
```

```
}
```

```
        u.setTestResult();
```

```
        u.display();
```

```
}
```

```
break;
```

```
case 2 : {
```

~~```
 System.out.println("Enter the student name : ");
```~~~~```
    Graduate g = new Graduate(s.nextInt());
```~~~~```
 System.out.print("Enter the subject number and marks of 9 sub ");
```~~~~```
    for (int i=0; i<9; i++)
```~~~~```
{
```~~~~```
        g.setTestScore(s.nextInt(), s.nextInt());
```~~~~```
}
```~~~~```
        g.setTestResult();
```~~~~```
 g.display();
```~~~~```
{
```~~~~```
break;
```~~~~```
default : System.out.println("Invalid choice ! ");
```~~~~```
{
```~~~~```
}
```~~

OUTPUT

1 Undergraduate Student

2. Graduate Student

Enter your choice :

1

Enter the student name : abc

Enter subject marks :

1 80

2 90

3 70

4 65

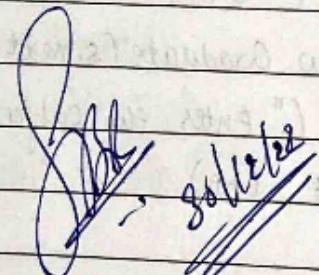
Student Name : abc

Student Marks :

80 90 70 65

Result :

Pass


Sohil, 20/12/22

PROGRAM 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;

class Negage extends Exception
{
    public String getMessage()
    {
        return "Age cannot be negative";
    }
}

class Invalidage extends Exception
{
    public String getMessage()
    {
        return "Son's age cannot be greater than Father's age";
    }
}

class Father
{
    int f;
    Father(int age) throws Negage
    {
        if(age<=0)
            throw new Negage();
        f = age;
    }
}

class Son extends Father
{
    int l;
    Son(int agef, int agel) throws Invalidage,Negage
    {
```

```

super(agef);
if(agef<=0)
    throw new Negage();
if(agef>agef)
    throw new Invalidage();
l = agef;
}
}

public class Exc
{
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("Enter Father's age and son's age");
            try {
                Son son = new Son(sc.nextInt(), sc.nextInt());
                System.out.println("Sons age: " + son.l + "\n" + "Father's age: " + son.f);
            } catch (Negage n) {
                System.out.println(n.getMessage());
            } catch (Invalidage i) {
                System.out.println(i.getMessage());
            }
        }
    }
}

```

OUTPUT:

```

Enter Father's age and son's age
10 0
Age cannot be negative
Enter Father's age and son's age
0 20
Age cannot be negative
Enter Father's age and son's age
20 30
Son's age cannot be greater than Father's age
Enter Father's age and son's age
30 10
Sons age: 10
Father's age: 30

```

```
import java.util.Scanner;  
  
class Negage extends Exception  
{  
    public String getMessage()  
    {  
        return "Age cannot be negative";  
    }  
}  
  
class Invalidage extends Exception  
{  
    public String getMessage()  
    {  
        return "Son's age cannot be greater than Father's age";  
    }  
}  
  
class Father  
{  
    int f;  
    Father(int age) throws Negage  
    {  
        if (age <= 0)  
            throw new Negage();  
        f = age;  
    }  
}  
  
class Son extends Father  
{  
    int l;  
    Son(int agef, int agel) throws Invalidage, Negage
```

{

super (agef);

if (agef <= 0)

throw new Negage();

if (agef > agef)

throw new Invalidage();

l = agef;

}

}

public class Exc

{

public static void main (String args [])

{

Scanner s = new Scanner (System.in) ;

System.out.println ("Enter father and son's age :");

try {

Son son = new Son (s.nextInt(), s.nextInt());

System.out.println ("Son's age :" + son.s + " \n " +

"Father's age :" + son.f);

}

catch (Negagen)

System.out.println (n.getMessage());

catch (Invalidage i)

System.out.println (i.getMessage());

}

}

Output

Enter Father and son's age :

34 12

Son's age : 12

Father's age : 34

Enter Father and son's age:

12 0

Age cannot be negative

Enter Father and son's age:

12 34

Son's age cannot be greater than father's age

Enter Father and son's age:

0 34

Age cannot be negative.

BB

27/11/23

PROGRAM 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
class Thread1 extends Thread {  
    public void run() {  
        while (true) {  
            try {  
                Thread.sleep(10000);  
                System.out.println("BMS College of Engineering");  
            } catch (Exception e) {  
                System.out.println(e);  
            }  
        }  
    }  
  
    class Thread2 extends Thread {  
        public void run() {  
            while (true) {  
                try {  
                    Thread.sleep(2000);  
                    System.out.println("CSE");  
                } catch (Exception e) {  
                    System.out.println(e);  
                }  
            }  
        }  
    }  
  
    class Threads {  
        public static void main(String args[]) {  
            Thread t1 = new Thread1();  
            Thread t2 = new Thread2();  
            t1.start();  
            t2.start();  
        }  
    }  
}
```

OUTPUT:

The screenshot shows a Windows desktop environment. At the top, there is a browser window titled "JAVA Lab Question on 13" with the URL "classroom.google.com/c/NTQwNjY3NDYyNjI0/a/NTgyOTI4ODMyMzc3/details". Below the browser is a command prompt window titled "JAVA PRORAMMING" with the path "C:\Windows\System32\cmd.exe - java Threads". The command prompt displays multiple lines of text, all reading "CSE", except for three instances of "BMS College of Engineering". The desktop taskbar at the bottom includes icons for File Explorer, Task View, Mail, Google Chrome, and File Explorer again. A system tray icon indicates "28°C Mostly sunny". The system clock shows "2:53 PM" and the date "1/13/2023".

```
C:\Windows\System32\cmd.exe - java Threads
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
```

Prog

```
class Thread1 extends Thread {  
    public void run() {  
        while (true) {  
            try {  
                Thread.sleep (10000);  
                System.out.println ("BMS College of Engineering");  
            } catch (Exception e) {  
                System.out.println (e);  
            }  
        }  
    }  
}
```

```
class Thread2 extends Thread {  
    public void run() {  
        while (true) {  
            try {  
                Thread.sleep (2000);  
                System.out.println ("CSE");  
            } catch (Exception e) {  
                System.out.println (e);  
            }  
        }  
    }  
}
```

```
class Threads {  
    public static void main (String args []) {  
        Thread t1 = new Thread1();  
        Thread t2 = new Thread2();  
        t1.start ();  
        t2.start ();  
    }  
}
```

Output

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

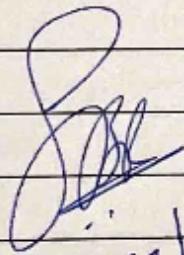
CSE

CSE

CSE

CSE

BMS College of Engineering


21/11/23