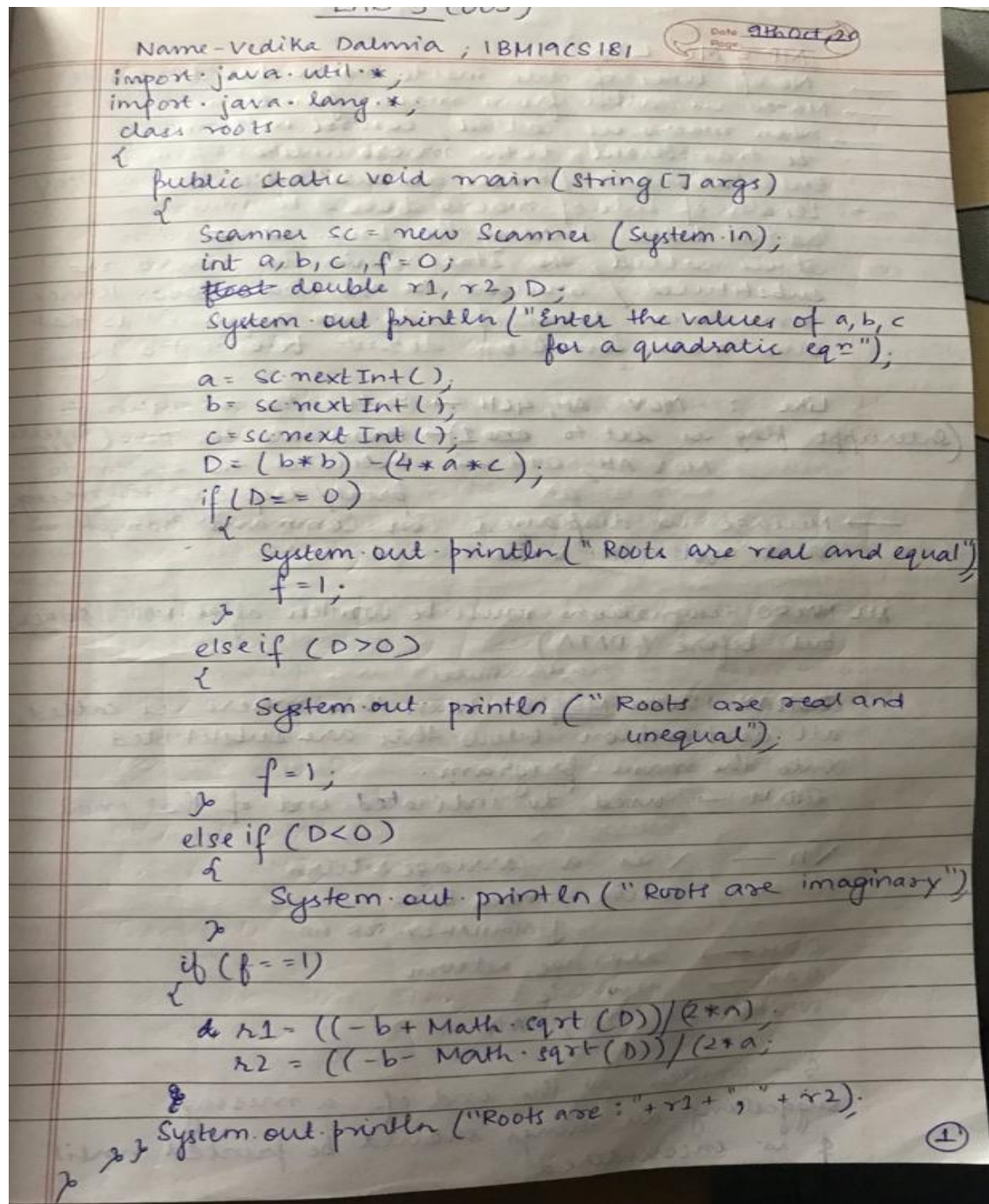


NAME-VEDIKA DALMIA; USN-1BM19CS181;SEC-3D;LAB BATCH-1

LAB PROGRAM-1:

Q.) 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 discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

WRITEUP:



The image shows a handwritten Java program on lined paper. At the top, the student's name 'Vedika Dalmia' and USN '1BM19CS181' are written. A date stamp '9/10/20' is visible. The code imports java.util.* and java.lang.* into a class named 'roots'. The main method uses a Scanner to take input for a, b, and c. It calculates the discriminant D = b*b - 4*a*c. If D is 0, it prints 'Roots are real and equal' and sets a flag f=1. If D is greater than 0, it prints 'Roots are real and unequal' and also sets f=1. If D is less than 0, it prints 'Roots are imaginary'. After the conditional blocks, it checks if f equals 1. If true, it calculates the two real roots r1 and r2 using the quadratic formula and prints them. The code is marked with several checkmarks and a circled '1' at the bottom right.

```
Name-Vedika Dalmia ; 1BM19CS181
Date 9/10/20
Page 1/1

import java.util.*;
import java.lang.*;
class roots
{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner (System.in);
        int a, b, c, f=0;
        float double r1, r2, D;
        System.out.println ("Enter the values of a, b, c
                             for a quadratic eq=");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        D = (b*b) - (4*a*c);
        if (D == 0)
        {
            System.out.println ("Roots are real and equal")
            f = 1;
        }
        else if (D > 0)
        {
            System.out.println ("Roots are real and
                                unequal");
            f = 1;
        }
        else if (D < 0)
        {
            System.out.println ("Roots are imaginary")
        }
        if (f == 1)
        {
            r1 = ((-b + Math.sqrt(D))/(2*a));
            r2 = ((-b - Math.sqrt(D))/(2*a));
            System.out.println ("Roots are : " + r1 + " , " + r2);
        }
    }
}
```

PROGRAM

```
import java.util.*;

class Roots

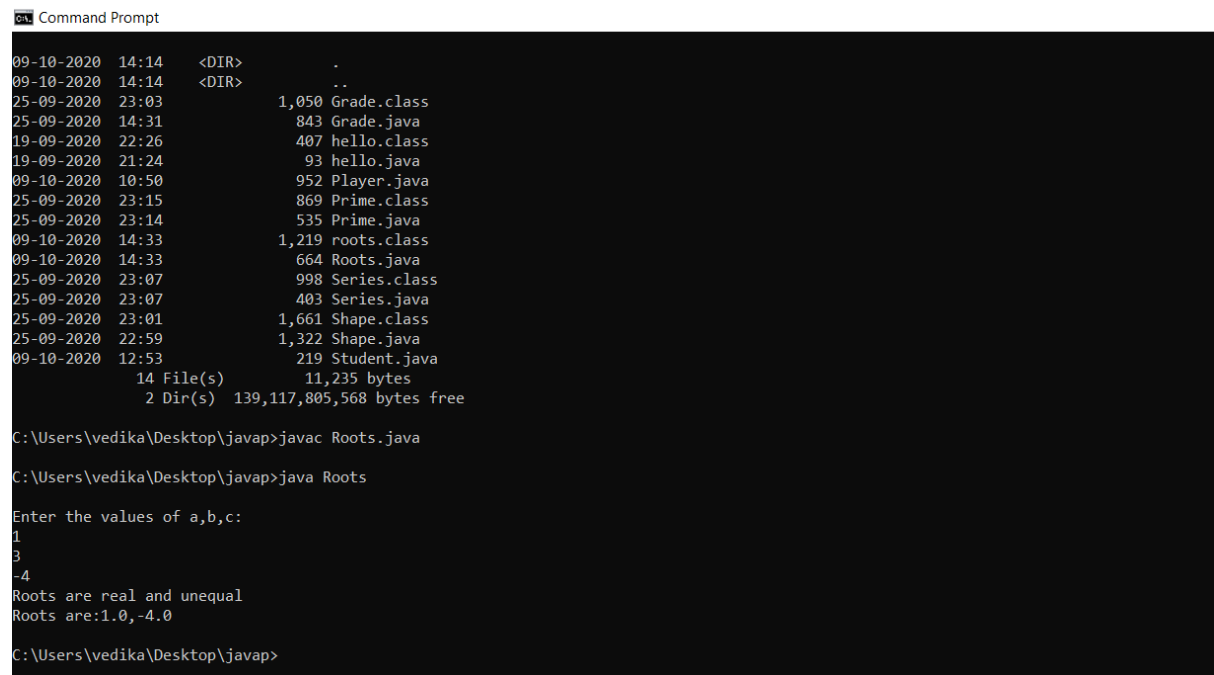
{
    public static void main(String[] args)
    {
        int a,b,c,f=0;
        double D;
        Scanner sc=new Scanner(System.in);
        System.out.println("\nEnter the values of a,b,c:");
        a=sc.nextInt();
        b=sc.nextInt();
        c=sc.nextInt();
        D=(b*b)-(4*a*c);
        if(D==0)
        {
            System.out.println("Roots are real and equal");
            f=1;
        }
        else if(D>0)
        {
            System.out.println("Roots are real and unequal");
            f=1;
        }
        else if(D<0)
        {
            System.out.println("Roots are imaginary");
```

```

    }
    if(f==1)
    {
        double r1=((-b+Math.sqrt(D))/(2*a));
        double r2=((-b-Math.sqrt(D))/(2*a));
        System.out.println("Roots are:"+r1+","+r2);
    }
}
}
}

```

OUTPUT SCREEN:1



```

C:\Users\vedika\Desktop\javap>javac Roots.java
C:\Users\vedika\Desktop\javap>java Roots
Enter the values of a,b,c:
1
3
-4
Roots are real and unequal
Roots are:1.0,-4.0
C:\Users\vedika\Desktop\javap>

```

The screenshot shows a Windows Command Prompt window with a black background and white text. It displays the execution of a Java program. The first part shows the directory listing of the current directory, which includes several Java class files. The second part shows the compilation of 'Roots.java' using 'javac'. The third part shows the execution of 'Roots' using 'java'. The program prompts the user to enter values for a, b, and c. The user enters 1, 3, and -4. The program then outputs 'Roots are real and unequal' and 'Roots are:1.0,-4.0'.

LAB 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.

WRITEUP:

Name - Vedika Dalma, USN - 18M19CS181
LAB-4 (003)

```
import java.util.*;
class Student
{
    String usn, name;
    int credits[], marks[];
    int n, tot = 0;
    double SGPA;

    Student()
    {
        tot = 0;
        SGPA = 0;
    }

    void input()
    {
        int i;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN and name of the student");
        usn = sc.nextLine();
        name = sc.nextLine();
        System.out.println("Enter the numbs of subjects");
        n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (i = 0; i < n; i++)
        {
            System.out.println("Enter the credits for subject : " + (i+1));
            credits[i] = sc.nextInt();
            tot = tot + credits[i];
        }
        for (i = 0; i < n; i++)
        {
            System.out.println("Enter the marks for student in sub : " + (i+1));
        }
    }
}
```



```

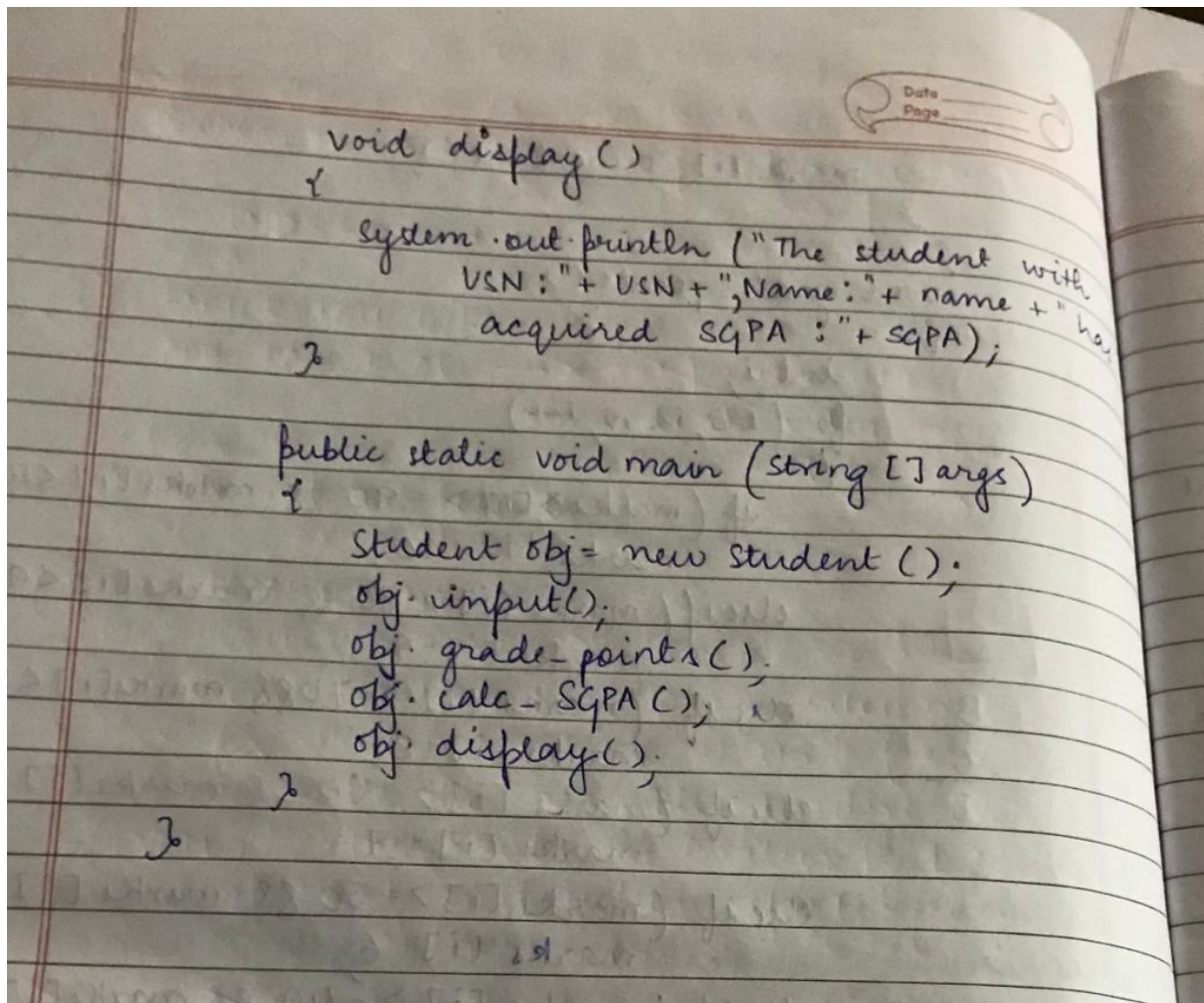
        marks[i] = cc.nextInt();
    }
}
void grade_points()
{
    int i;
    for (i=0; i<n; i++)
    {
        if (marks[i] >= 90 && marks[i] <= 100)
            marks[i] = 10;
        else if (marks[i] >= 80 && marks[i] < 90)
            marks[i] = 9;
        else if (marks[i] >= 70 && marks[i] < 80)
            marks[i] = 8;
        else if (marks[i] >= 60 && marks[i] < 70)
            marks[i] = 7;
        else if (marks[i] >= 50 && marks[i] < 60)
            marks[i] = 6;
        else if (marks[i] >= 40 && marks[i] < 50)
            marks[i] = 4;
        else if (marks[i] < 40)
            marks[i] = 0;
    }
}

```

```

}
void calc_grade SQPA()
{
    int i;
    for (i=0; i<n; i++)
    {
        SQPA = SQPA + (credits[i] * marks[i]);
    }
    SQPA = SQPA / tot;
}

```



PROGRAM

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

```
class Student2
```

```
{
```

```
    String USN;
```

```
    String name;
```

```
    int credits[];
```

```
    int marks[];
```

```
    int n,tot;
```

```
double SGPA;
```

```
Student2()
```

```
{
```

```
tot=0;
```

```
SGPA=0;
```

```
}
```

```
void input()
```

```
{
```

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

```
System.out.println("Enter the USN and the name of the student");
```

```
USN=sc.nextLine();
```

```
name=sc.nextLine();
```

```
System.out.println("Enter the number of subjects");
```

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

```
credits=new int[n];
```

```
marks =new int[n];
```

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

```
{
```

```
System.out.println("Enter the credits for subject: "+(i+1));
```

```
credits[i]=sc.nextInt();
```

```
tot=tot+credits[i];
```

```
}
```

```
for(int i=0;i<n;i++)
{
    System.out.println("Enter the marks of the student for subject: "+(i+1));
    marks[i]=sc.nextInt();
}
}

void grade_points()
{
    int i;
    for(i=0;i<n;i++)
    {
        if(marks[i]>=90 && marks[i]<100)
        {
            marks[i]=10;
        }
        else if(marks[i]>=80 && marks[i]<90)
        {
            marks[i]=9;
        }
        else if(marks[i]>=70 && marks[i]<80)
        {
            marks[i]=8;
        }
    }
}
```



```
    else if(marks[i]>=60 && marks[i]<70)
    {
        marks[i]=7;
    }
    else if(marks[i]>=50 && marks[i]<60)
    {
        marks[i]=6;
    }
    else if(marks[i]>=40 && marks[i]<50)
    {
        marks[i]=4;
    }
    else if(marks[i]<40)
    {
        marks[i]=0;
    }
}

void calc_SGPA()
{
    int i;
    for(i=0;i<n;i++)
    {
```

```
    SGPA=SGPA+(credits[i]*marks[i]);  
  
    }  
  
    SGPA=SGPA/tot;  
  
    }  
  
void display()  
  
{  
  
    System.out.println("The student with USN:"+USN+", Name:"+name+" has  
    SGPA:"+SGPA);  
  
    }  
  
public static void main(String[] args)  
  
{  
  
  
  
  
    Student2 obj=new Student2();  
  
    obj.input();  
  
    obj.grade_points();  
  
    obj.calc_SGPA();  
  
    obj.display();  
  
    }  
  
    }
```

OUTPUT SCREEN:2

Command Prompt

```
C:\Users\vedika\Desktop\javap>java Student2
Enter the USN and the name of the student
1BM19CS181
VEDIKA DALMIA
Enter the number of subjects
5
Enter the credits for subject:1
5
Enter the credits for subject:2
4
Enter the credits for subject:3
4
Enter the credits for subject:4
4
Enter the credits for subject:5
3
Enter the marks of the student for subject:1
69
Enter the marks of the student for subject:2
76
Enter the marks of the student for subject:3
82
Enter the marks of the student for subject:4
80
Enter the marks of the student for subject:5
80
The student with USN:1BM19CS181, Name:VEDIKA DALMIA has SGPA:8.3
C:\Users\vedika\Desktop\javap>
```

LAB PROGRAM 3:

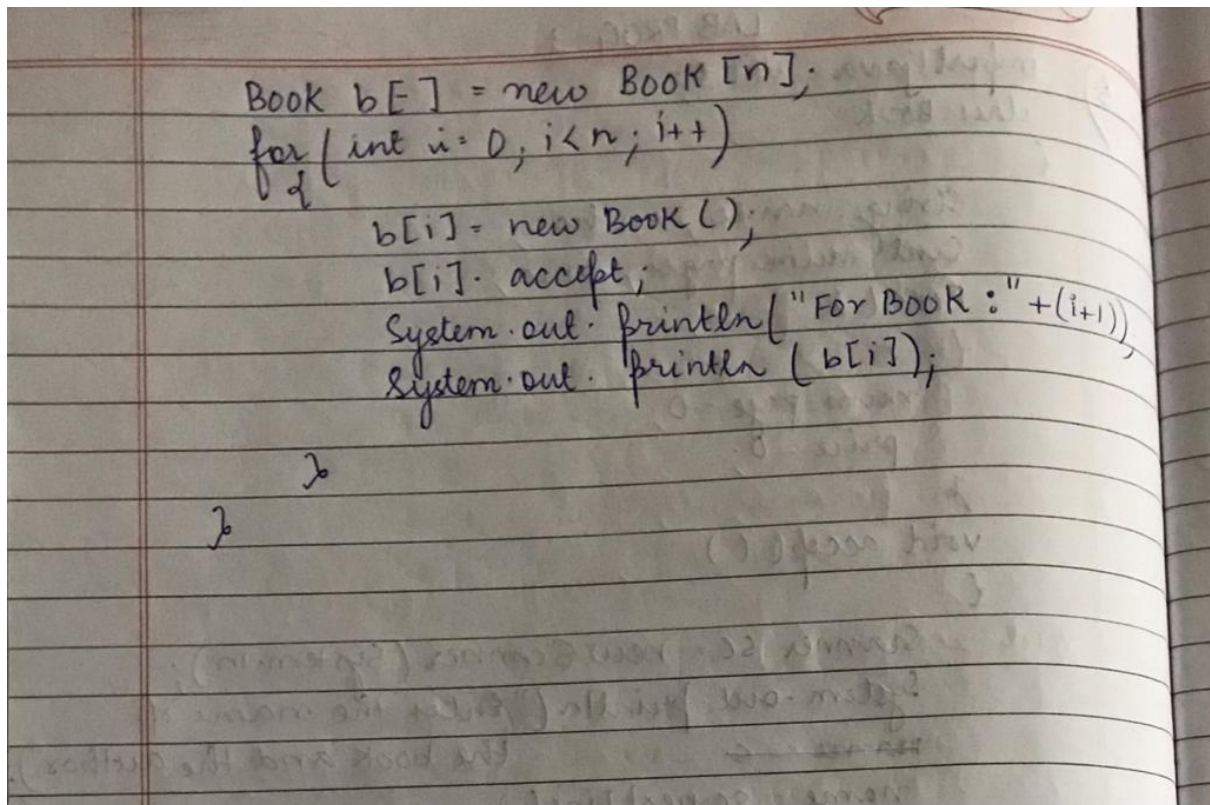
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program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

WRITEUP:

LAB PROG-3

Date: 6th Nov 20
Page: _____

```
8) import java.util.*;
class Book
{
    String name, author;
    int num_pages, price;
    Book()
    {
        num_page = 0;
        price = 0;
    }
    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of-  
name = sc the book and the author");
        name = sc.nextLine();
        author = sc.nextLine();
        System.out.println("Enter the price of the  
books and the number  
of pages in it");
        price = sc.nextInt();
        num_page = sc.nextInt();
    }
    public String toString()
    {
        return ("Name of the book : " + name +  
"In Author : " + author + "\n Price : " +  
price + "\n No. of Pages : " + num_pages);
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        Book b = new Book
        System.out.print("Enter value of n\n");
        int n = sc.nextInt();
    }
}
```



PROGRAM

```
import java.util.Scanner;  
  
class Book  
{  
    private String name, author;  
    private double price;  
    private int num_pages;  
  
    Book()  
{  
    name="A";  
    author="BCD";
```



```
        price=340.0;
        num_pages=500;
    }

    void Input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the book name: ");
        name=sc.nextLine();
        System.out.println("Enter the author name: ");
        author=sc.nextLine();
        System.out.println("Enter the the no.of pages: ");
        num_pages=sc.nextInt();
        System.out.println("Enter the price: ");
        price=sc.nextDouble();
    }

    public String toString()
    {
        String temp="Book name: "+name+"\nAuthor name: "+author+"\nNo.of
pages: "+num_pages+"\nPrice: "+price+"\n";
        return(temp);
    }
```

```
}  
  
class B  
  
{  
  
    public static void main(String args[])  
  
    {  
  
int i,n;  
  
Scanner sc=new Scanner(System.in);  
  
System.out.print("Enter the number of books: ");  
  
n=sc.nextInt();  
  
Book[] obj=new Book[n];  
  
for(i=0;i<n;i++)  
  
{  
  
    obj[i]=new Book();  
  
}  
  
System.out.println("\t\t**Enter Book Details**");  
  
    for(i=0;i<n;i++)  
  
{  
  
        System.out.println("\nBook "+(i+1)+";");  
  
        obj[i].Input();  
  
}  
  
System.out.println("Book Details:");  
  
    for(i=0;i<n;i++)  
  
{
```

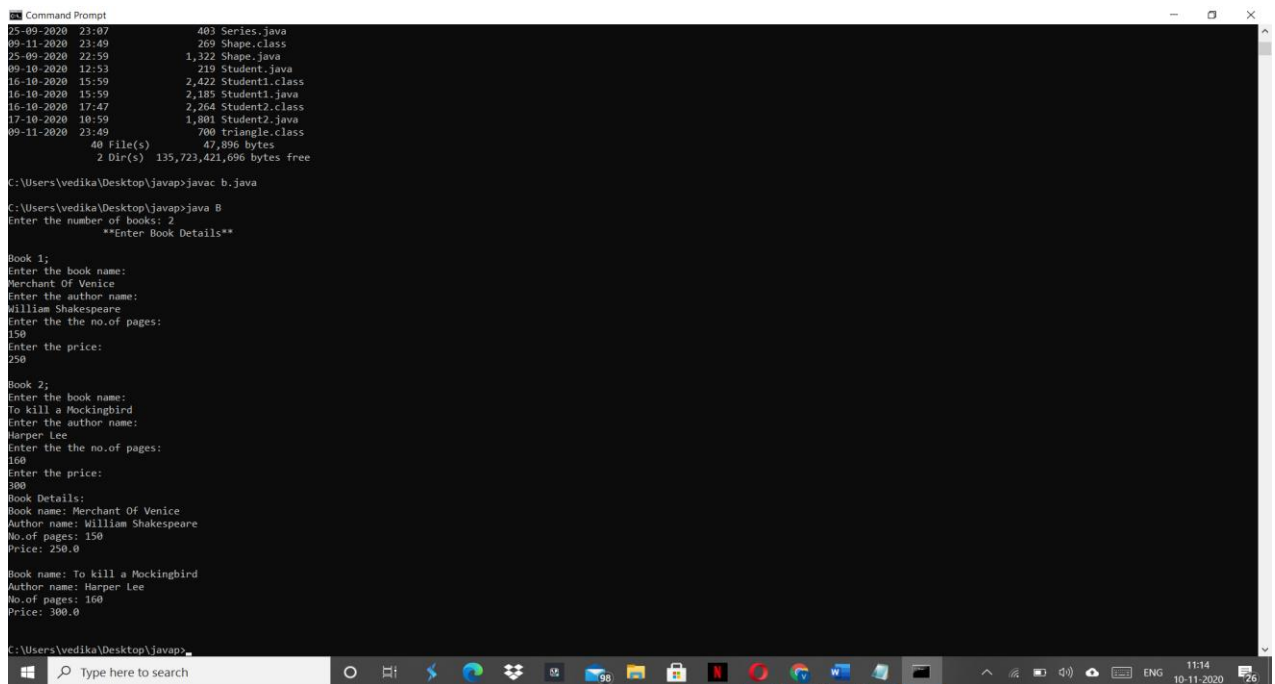
```
System.out.println(obj[i]);
```

```
}
```

```
}
```

```
}
```

OUTPUT SCREEN:3



```
Command Prompt
25-09-2020 23:47      403 Series.java
09-11-2020 23:49      269 Shape.class
25-09-2020 22:59      1,322 Shape.java
09-10-2020 12:53      219 Student.java
16-10-2020 15:59      2,422 Student1.class
16-10-2020 15:59      2,185 Student1.java
16-10-2020 17:47      2,264 Student2.class
17-10-2020 10:59      1,801 Student2.java
09-11-2020 23:49      700 triangle.class
      48 File(s)      47,896 bytes
      2 Dir(s)  135,723,421,696 bytes free

C:\Users\vedika\Desktop\javap>javac b.java

C:\Users\vedika\Desktop\javap>java B
Enter the number of books: 2
**Enter Book Details**

Book 1:
Enter the book name:
Merchant Of Venice
Enter the author name:
William Shakespeare
Enter the the no.of pages:
150
Enter the price:
250

Book 2:
Enter the book name:
To kill a Mockingbird
Enter the author name:
Harper Lee
Enter the the no.of pages:
160
Enter the price:
300

Book Details:
Book name: Merchant Of Venice
Author name: William Shakespeare
No.of pages: 150
Price: 250.0

Book name: To kill a Mockingbird
Author name: Harper Lee
No.of pages: 160
Price: 300.0

C:\Users\vedika\Desktop\javap>
```

LAB 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.

WRITEUP:

LAB PROG-5

Name-Vedika Dalmia; USN-1BM19CS187

Date 09/11/20

5) abstract class Shape

```
{
    int a=3;
    int b=4;
    abstract public void printArea(),
}
```

```
class rectangle extends Shape
{
    public int area_rect;

    public void printArea()
    {
        area_rect = a * b;
        System.out.println("Area of Rect : " + area_rect);
    }
}
```

```
class triangle extends Shape
{
    int area_circle;
    public void printArea()
    {
        area_circle = (int) (3.14 * a * a);
        System.out.println("Area of Circle : " + area_circle);
    }
}
```

```
class AreaShape
{
    public static void main (String args[])
    {
        rectangle rec = new rectangle();
        rec.printArea();
        triangle tri = new triangle();
        tri.printArea();
        circle cir = new circle();
        cir.printArea();
    }
}
```

PROGRAM

```
abstract class Shape

{

int a=3;

int b=4;

abstract public void print_area();

}

class rectangle extends Shape

{

public int area_rect;


public void print_area()

{

area_rect=a*b;

System.out.println("The area of rectangle is: "+area_rect);

}

}

class triangle extends Shape

{

int area_tri;


public void print_area()
```



```
{  
    area_tri=(int) (0.5*a*b);  
    System.out.println("The area of triangle is: "+area_tri);  
}  
}  
  
class circle extends Shape  
{  
    int area_circle;  
  
    public void print_area()  
    {  
        area_circle=(int) (3.14*a*a);  
        System.out.println("The area of circle is: "+area_circle);  
    }  
}  
  
class abs{  
    public static void main(String[] args){  
        rectangle rec = new rectangle();  
        rec.print_area();  
        triangle tri = new triangle();  
        tri.print_area();  
        circle cir = new circle();  
        cir.print_area();  
    }  
}
```

}

}

OUTPUT SCREEN:4

```
Command Prompt
02-11-2020 19:42      1,543 Matrix.class
02-11-2020 19:41      1,261 Matrix.java
09-10-2020 10:50       952 Player.java
10-10-2020 18:01      1,207 pos_neg.class
10-10-2020 18:01       905 pos_neg.java
25-09-2020 23:15       869 Prime.class
25-09-2020 23:14       535 Prime.java
09-10-2020 14:35      1,219 roots.class
09-10-2020 14:33       664 Roots.java
07-11-2020 15:07      1,207 Sav_rect.class
25-09-2020 23:07       998 Series.class
25-09-2020 23:07       403 Series.java
25-09-2020 23:01      1,661 Shape.class
25-09-2020 22:59      1,322 Shape.java
09-10-2020 12:53       219 Student.java
16-10-2020 15:59      2,422 Student1.class
16-10-2020 15:59      2,185 Student1.java
16-10-2020 17:47      2,264 Student2.class
17-10-2020 10:59      1,801 Student2.java
      35 File(s)      45,560 bytes
      2 Dir(s)  132,964,204,544 bytes free

C:\Users\vedika\Desktop\javap>javac abs.java

C:\Users\vedika\Desktop\javap>java abs
The area of rectangle is: 12
The area of triangle is: 6
The area of circle is: 28

C:\Users\vedika\Desktop\javap>
```

LAB 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 Curr-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 Check for the minimum balance, impose penalty if necessary and update the balance.

WRITEUP:

LAB PROG-4
Name - Vedika Dalmia, USN - IBM19CS181 Date - 06/Nov/20
Page -
4.)

```
import java.util.*;
class Account
{
    private String name;
    private long account_num;
    private int account_type;
    double balance;
    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter account holders name");
        name = sc.next();
        System.out.println("Enter acct no.");
        account_num = sc.nextLong();
        System.out.println("Account type : Enter : \n 1. Savings Acct \n 2. Current acct");
        account_type = sc.nextInt();
    }
    void get_data()
    {
        System.out.println("Acct holders name:" + name);
        System.out.println("Acct Numb.:" + account_num);
    }
    int return_account_type()
    {
        return account_type;
    }
}
```

class savings extends Account

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

double amount;

void get_sav_balance ()

{
System.out.println("Enter amount you wish
to deposit");

amount = sc.nextDouble();

balance = balance + amount;

}

void display_sav_balance ()

{

System.out.println("balance = " + balance);

}

void compute_sav_interest ()

{

System.out.println("Calculating Comp Interest

System.out.println("Enter rate");

float rate = sc.nextFloat();

System.out.println("Enter time in years");

int time = sc.nextInt();

System.out.println("Enter principle");

float P = sc.nextFloat();

float CI = (float) ((P * (Math.pow(1 + rate / 12, 12 * time))) - P);

System.out.println("CI = " + CI);

balance = balance + CI;

System.out.println("After adding interest
balance = " + balance);

}

void withdrawl_sav ()

{ double amount;

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

amount = sc.nextDouble();

balance = balance - amount;

}


```

class current extends Account
{
    Scanner sc = new Scanner(System.in);
    double amount;
    final double min_balance = 5000;
    void get_cur_balance()
    {
        System.out.println("Enter the amount to be
        placed in your current acct");
        amount = sc.nextDouble();
        balance = balance + amount;
    }
    void display_cur_balance()
    {
        System.out.println("Balance = " + balance);
    }
    void compute_cur_service_charges()
    {
        if (balance < min_balance)
        {
            System.out.println("Service tax of Rs.150
            shall be charged to your acct");
            balance = balance - 150;
        }
        else
        {
            System.out.println("Min balance is
            maintained");
        }
    }
    void withdraw_cur()
    {
        System.out.println("Enter the amt. to be
        withdrawn");
        amount = sc.nextDouble();
        balance = balance - amount;
    }
}

```


class Bank

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

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

```
        int type;
```

```
        System.out.println ("Enter bank details");
```

```
        Account acc = new Account ();
```

```
        acc.input ();
```

```
        type = acc.return_acct_type ();
```

```
        if (type == 1)
```

```
{
```

```
            System.out.println ("Savings Acct");
```

```
            acc.get_data ();
```

```
            Savings sav = new Savings ();
```

```
            sav.get_sav_balance ();
```

```
            sav.display_sav_balance ();
```

```
            System.out.print ("Calculating  
                             interest");
```

```
            sav.compute_sav_interest ();
```

```
            sav.display_sav_balance ();
```

```
            sav.withdrawal_sav ();
```

```
            sav.display_sav_balance ();
```

```
}
```

```
        else if (type == 2)
```

```
{
```

```
            System.out.println ("Current Account");
```

```
            acc.get_data ();
```

```
            Current cur = new Current ();
```

```
            cur.get_cur_balance ();
```

```
            cur.display_cur_balance ();
```

```
            cur.compute_cur_service_charges ();
```

```
            cur.display_cur_balance ();
```

```
            cur.withdrawal_cur ();
```

```
            cur.display_cur_balance ();
```

```
}
```

6

PROGRAM

```
import java.util.Scanner;

class Account

{

private String name;

private long account_number;

private int account_type;

double balance;

void Input()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter Account Holder Name");

name = sc.nextLine();

System.out.println("Enter the account Number");

account_number=sc.nextLong();

System.out.println("Choose the account type:\n1.savings account\n2.current account");

account_type=sc.nextInt();

}

void get_data(){

System.out.println("Account Holder: "+name);

System.out.println("Account Number: "+account_number);

}
```

```
int return_account_type()
{
return account_type;
}
}

class savings extends Account
{
Scanner sc=new Scanner(System.in);

double amount;

void get_sav_balance()
{
System.out.println("Enter the Amount to be placed in your Savings Account");
amount=sc.nextDouble();

balance+=amount;
}

void display_sav_blnce()
{
System.out.println("balance= "+balance);
}

void compute_sav_interest()
{
System.out.println("\n***Calculating Compound Interest***");

System.out.println("Enter annual interest rate: ");
```

```
float rate = sc.nextFloat();

System.out.println("Enter time in years: ");

float time = sc.nextFloat();

System.out.println("Enter principle: ");

float principle = sc.nextFloat();

float CI = (float)((principle*(Math.pow((1 + rate / (12*100)),(12*time))))-principle);

System.out.println("The Compound Interest is: " +CI);

balance = balance+CI;

System.out.println("Balance after adding Interest: "+balance);

}

void withdrawl_sav()

{

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

amount = sc.nextDouble();

balance=balance-amount;

}

}

class current extends Account

{

Scanner sc = new Scanner(System.in);

double amount;

final double min_balance=500;

void get_cur_balance()
```

```
{  
System.out.println("Enter the amount to be placed in your current account");  
amount=sc.nextDouble();  
balance+=amount;  
}  
void display_cur_blnce()  
{  
System.out.println("Balance = "+balance);  
}  
void compute_cur_service_charges()  
{  
if(balance<min_balance)  
{  
System.out.println("service tax of rs.100 shall be levied");  
balance=balance-100;  
}  
else  
{  
System.out.println("Minimum balance is Maintained");  
}  
}  
void withdrawl_cur()  
{
```



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

```
amount=sc.nextDouble();
```

```
balance=balance-amount;
```

```
}
```

```
}
```

```
class BankF
```

```
{
```

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

```
{
```

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

```
int type;
```

```
System.out.println("Enter the bank details");
```

```
Account acc=new Account();
```

```
acc.Input();
```

```
type=acc.return_account_type();
```

```
if (type==1)
```

```
{
```

```
System.out.println("SAVINGS ACCOUNT");
```

```
acc.get_data();
```

```
savings sav = new savings();
```

```
sav.get_sav_balance();
```

```
sav.display_sav_blnce();
```

```
System.out.println("Calculating Interest:\n");
```

```
sav.compute_sav_interest();

sav.display_sav_blnce();

sav.withdrawl_sav();

sav.display_sav_blnce();

}

if(type==2)

{

System.out.println("CURRENT ACCOUNT");

acc.get_data();

current cur=new current();

cur.get_cur_balance();

cur.display_cur_blnce();

cur.compute_cur_service_charges();

cur.display_cur_blnce();

cur.withdrawl_cur();

cur.display_cur_blnce();

}

}

}
```

OUTPUT SCREEN:5

```
Command Prompt
07-11-2020 15:07      1,207 Sav_acct.class
25-09-2020 23:07      998 Series.class
25-09-2020 23:07      403 Series.java
09-11-2020 23:40      269 Shape.class
25-09-2020 22:59      1,322 Shape.java
09-10-2020 12:53      219 Student.java
16-10-2020 15:59      2,422 Student1.class
16-10-2020 15:59      2,185 Student1.java
16-10-2020 17:47      2,264 Student2.class
17-10-2020 10:59      1,801 Student2.java
09-11-2020 23:49      700 triangle.class
43 File(s)           53,616 bytes
2 Dir(s)            135,718,027,264 bytes free

C:\Users\vedika\Desktop\javap>javac BankF.java

C:\Users\vedika\Desktop\javap>java BankF
Enter the bank details
Enter Account Holder Name
ARUN
Enter the account Number
12346
Choose the account type:
1.savings account
2.current account
1
SWIMES ACCOUNT
Account Holder: ARUN
Account Number: 12346
Enter the Amount to be placed in your Savings Account
12000
balance- 12000.0
calculating Interest:

***Calculating Compound Interest***
Enter annual Interest rate:
3
Enter time in years:
3
Enter principle:
12000
The Compound Interest is: 1128.6438
Balance after adding Interest: 13128.643798828125
balance- 13128.643798828125
Enter the amount to be withdrawn
1200
balance- 11928.643798828125

C:\Users\vedika\Desktop\javap>
```

Lab Program: 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses. Below is two programs of class Student and class Internals. Both belong to a package named CIE. Class Internals extends from class Student.

WRITEUP:

LAB PROG - 6

Name - Vedika Dalmia, USN - IBM19CS181

```
package CIE;
import java.util.Scanner;
public class Student
{
    public String name;
    public String USN;
    public int sem;
    public void display()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Name:");
        name = sc.nextLine();
        System.out.println("USN:");
        USN = sc.nextLine();
        System.out.println("Semester:");
        sem = sc.nextInt();
    }
}
```

```
package CIE;
import java.util.*;
public class Internals extends Student
{
    public double ciem[5];
    public void display()
    {
        ciem = new double[5];
        Scanner sc = new Scanner(System.in);
        System.out.println("CIE marks:");
        for (i = 0; i < 5; i++)
        {
            ciem[i] = sc.nextDouble();
        }
    }
}
```

```

package SEE;
import java.util.*;
import CIE.*;
public class Extends CIE Student
{
    public double seem[];
    public void display()
    {
        seem = new double[5];
        Scanner sc = new Scanner(System.in);
        System.out.println("SEE Marks for 5 sub:");
        for(i=0; i<5; i++)
        {
            seem[i] = sc.nextDouble();
        }
    }
}

```

DRIVER CLASS

```

import CIE.*;
import SEE.*;
import java.util.*;
public class Main
{
    public static void main(String args[])
    {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of student");
        n = sc.nextInt();
        CIE.Students st[] = new CIE.Students[n];
        CIE.Internals in[] = new CIE.Internals[n];
        SEE.Externals e[] = new SEE.Externals[n];
        for(i=0; i<n; i++)
        {

```

```

s[i] = new CIE.Student();
in[i] = new CIE.Internals();
e[i] = new GEE.Externals();
s[i].display();
in[i].display();
e[i].display();
System.out.println("Total marks of student " +
s[i].name + in 5 subjects are :");
for (j=0; j<5; j++)
{
    System.out.println(in[i].ciem[j] + (e[i].seem[j]));
}
}
}
}

```

PROGRAM

File with name Student.java

```

package CIE;
import java.util.Scanner;
public class Student
{
    public String name;
    public String usn;
    public int sem;
    public void display()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Name:");
        name=s.next();
        System.out.println("USN:");
        usn=s.next();
        System.out.println("Semester:");
        sem=s.nextInt();
    }
}

```

```
}
```

File with name Internals.java

```
package CIE;
import java.util.Scanner;

public class Internals extends Student

{
public double ciem[];
public void display()
{
ciem=new double[5];
Scanner t=new Scanner(System.in);
System.out.println("CIE Marks for 5 subjects(out of 50):");
for(int i=0;i<5;i++)
ciem[i]=t.nextDouble();
}
}
```

File with name Externals.java

```
package SEE;
import CIE.*;
import java.util.*;

public class Externals extends CIE.Student
{
public double seem[];
public void display()
{
seem=new double[5];
Scanner s=new Scanner(System.in);
System.out.println("SEE Marks for 5 subjects(out of 100):");

for(int i=0;i<5;i++)
seem[i]=s.nextDouble();
}
}
```


File with name Main.java

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main
{
    public static void main(String args[])
    {
        int n;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the number of students:");
        n=s.nextInt();
        CIE.Student st[]=new CIE.Student[n];
        CIE.Internals in[]=new CIE.Internals[n];
        SEE.Externals e[]=new SEE.Externals[n];
        for(int i=0;i<n;i++)
        {
            st[i]=new CIE.Student();
            in[i]=new CIE.Internals();
            e[i]=new SEE.Externals();

            st[i].display();
            in[i].display();
            e[i].display();
            System.out.println("Total marks of student"+st[i].name+" in 5
            subjects are:");
            for(int j=0;j<5;j++)
            {
                System.out.println(in[i].ciem[j]+(e[i].seem[j]/2));
            }
        }
    }
}
```

OUTPUT SCREEN:6

Command Prompt - java Main

```
C:\Users\vedika\Desktop\javap\PACKAGEPROG>java Main
Enter the number of students:
2
Name:
Vedika
USN:
1BM19CS181
Semester:
3
CIE Marks for 5 subjects(out of 50):
35
36
39
40
38
SEE Marks for 5 subjects(out of 100):
89
78
90
80
97
Total marks of studentVedika in 5 subjects are:
79.5
75.0
84.0
80.0
86.5
Name:
_
```

Lab Program: 7

Write a program to demonstrate generics with multiple object parameters.

WRITEUP:

LAB PROG-7

Name-Vedika Dalmia, USN-IBM19CS181

```
class mygen <a,b>
{
    a obj1;
    b obj2;

    mygen (a obj1, b obj2)
    {
        this.obj1 = obj1;
        this.obj2 = obj2;
    }

    void Display ()
    {
        System.out.println(obj1);
        System.out.println(obj2);
    }
}

public class genericsMain
{
    public static void main (String args[])
    {
        mygen <String, Integer> myg1 = new mygen <String,
        Integer> ("ved", 27);

        mygen <Character, Double> myg2 = new mygen <Character,
        Double> ('V', 27.77);

        myg1.Display ();
        myg2.Display ();
    }
}
```

PROGRAM

```
class myGen<a,b>{  
    a obj1;  
    b obj2;  
    myGen(a obj1, b obj2){  
        this.obj1 = obj1;  
        this.obj2 = obj2;  
    }  
}
```

```
void Display(){  
    System.out.println(obj1);  
    System.out.println(obj2);  
}  
}
```

```
public class Genericsmain{  
    public static void main(String args[]){  
        myGen<String,Integer>myG1 = new  
        myGen<String,Integer>("Vedika",27);  
        myGen<Character,Double>myG2           =           new  
        myGen<Character,Double>('V',27.77);  
        myG1.Display();  
        myG2.Display();  
    }  
}
```

OUTPUT SCREEN:7

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1316]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\vedika>cd desktop

C:\Users\vedika\Desktop>cd javap

C:\Users\vedika\Desktop\javap>javac Genericsmain.java

C:\Users\vedika\Desktop\javap>java Genericsmain
Vedika
27
V
27.77

C:\Users\vedika\Desktop\javap>_
```

Lab Program: 8

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 takes both father and son's age and throws an exception if son's age is >= father's age.

WRITEUP

LAB PROG-8
Name - Vedika Dalmia ; USN - IBM19CS181

```
import java.util.*;
class WrongAge extends Exception
{
    public WrongAge(String s)
    {
        super(s);
    }
}

class Father
{
    int fatherAge;
    Father (int fAge) throws WrongAge
    {
        if (fAge < 0)
        {
            throw new WrongAge ("Father's age is less than zero");
        }
        else
        {
            fatherAge = fAge;
        }
    }
}

class Son extends Father
{
    int sonAge;
    Son (int fAge, int sAge) throws WrongAge
    {
        super (fAge);
        sonAge = sAge;
        if (sAge >= fAge)
        {
            throw new WrongAge ("Son's age is equal or greater than father's age");
        }
    }
}
```


or greater than father's age");

```
}  
}  
void Display()  
{  
    System.out.println("Father's age : "+ fatherAge);  
}
```

```
}  
  
public class exp
```

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

```
{  
    int fAge, sAge;
```

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

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

```
    fAge = sc.nextInt();
```

```
    System.out.println ("Enter son's Age");
```

```
    sAge = sc.nextInt();
```

```
    try
```

```
{  
        Son son = new Son (fAge, sAge);
```

```
        son.display();
```

```
    } catch (WrongAge err)
```

```
{  
        System.out.println ("Exception : "+ err);  
    }
```

```
}  
}
```


PROGRAM

```
import java.util.Scanner;

class WrongAge extends Exception
{
    public WrongAge(String s)
    {
        super(s);
    }
}

class Father
{
    int fatherAge;

    Father(int fAge) throws WrongAge{
        if(fAge <=0)
        {
            throw new WrongAge("Father's age is less than 0");
        }
        else
        {
            this.fatherAge = fAge;
        }
    }
}
```

```
class Son extends Father
{
int sonAge;
Son(int fAge, int sAge) throws WrongAge{
super(fAge);
sonAge=sAge;
if(sAge >= fAge){

throw new WrongAge("Sons's age is equal to or greater than father's
age");
}

}
void Display(){
System.out.println("Father's age: "+fatherAge);
System.out.println("Son's age: "+sonAge);
}
}
```

```
class lab8 {
public static void main(String[] args)
{
int fAge,sAge;
Scanner sc = new Scanner(System.in);
System.out.println("Enter father's age: ");
fAge = sc.nextInt();
```

```

System.out.println("Enter sons's age: ");
sAge = sc.nextInt();
try{
Son son = new Son(fAge, sAge);
son.Display();
}catch(WrongAge err)
{
System.out.println("Exception " + err);
}
}
}

```

OUTPUT SCREEN-8

```

Command Prompt
C:\Users\vedika>cd desktop
C:\Users\vedika\Desktop>cd javap
C:\Users\vedika\Desktop\javap>javac lab8.java
C:\Users\vedika\Desktop\javap>java lab8
Enter father's age:
40
Enter sons's age:
12
Father's age: 40
Son's age: 12

C:\Users\vedika\Desktop\javap>java lab8
Enter father's age:
0
Enter sons's age:
12
Exception WrongAge: Father's age is less than 0

C:\Users\vedika\Desktop\javap>java lab8
Enter father's age:
10
Enter sons's age:
12
Exception WrongAge: Sons's age is equal to or greater than father's age

C:\Users\vedika\Desktop\javap>

```

Lab Program: 9

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.

WRITEUP

LAB PROG-9

Name - Vedika Dalmia, USN - IBM19C518/

```
class Thread1 implements Runnable
{
    String name;
    Thread t;
    Thread1 (String threadname)
    {
        name = threadname;
        t = new Thread (this, name);
        t.start();
    }

    public void run()
    {
        try {
            for (int i = 5; i > 0; i--)
            {
                System.out.println("Thread1: " + name);
                Thread.sleep(10000);
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(name + "Interrupted");
        }
        System.out.println(name + "exiting");
    }
}

class Thread2 implements Runnable
{
    String name;
    Thread t;
    Thread2 (String threadname)
    {
        name = threadname;
        t = new Thread (this, name);
        t.start();
    }
}
```

```

public void run()
{
    try {
        for (int i=5; i>0; i--)
        {
            System.out.println("Thread 2:" + name);
            Thread.sleep(2000);
        }
    }
    catch (InterruptedException e)
    {
        System.out.println(name + "Interrupted");
    }
    System.out.println(name + "Exiting");
}

}

class lab9 {
    public static void main (String args[])
    {
        new Thread1("BMS COLLEGE OF ENGINEERING");
        new Thread2("CSE");
        try
        {
            Thread.sleep(100000);
            System.out.println("Main Thread is awake");
        }
        catch (InterruptedException e)
        {
            System.out.println("Main thread Interrupted");
        }
        System.out.println("Main Thread Exiting")
    }
}

```

PROGRAM

```
class Thread1 implements Runnable
{
String name;
Thread t;
Thread1(String threadname)
{
name=threadname;
t=new Thread(this,name);
t.start();
}
public void run()
{
try{
for(int i=5;i>0;i--)

{
System.out.println("Thread1 "+name);
Thread.sleep(10000);
}
}
catch(InterruptedException e)
{
System.out.println(name+"Interrupted");
}
```



```
System.out.println(name+"exiting");  
}  
}
```

```
class Thread2 implements Runnable  
{  
String name;  
Thread t1;  
Thread2(String threadname)  
{  
name=threadname;  
t1=new Thread(this,name);  
t1.start();  
}  
public void run()  
{  
  
try{  
for(int i=5;i>0;i--)  
{  
System.out.println("Thread2"+name);  
Thread.sleep(2000);  
}  
}  
catch(InterruptedException e)  
{
```

```
System.out.println(name+"Interrupted");
}
System.out.println(name+"exiting");
}
}
class lab9{
public static void main(String args[])
{
new Thread1("BMS COLLEGE OF ENGINEERING");
new Thread2("CSE");
try{
Thread.sleep(100000);
System.out.println("Main thread is awake");
}
catch(InterruptedException e)
{

System.out.println("Main thread Interrupted");
}
System.out.println("Main thread exiting");
}
}
```

OUTPUT SCREEN:9

```
C:\Users\vedika\Desktop>cd javap
C:\Users\vedika\Desktop\javap>javac lab9.java
C:\Users\vedika\Desktop\javap>java lab9
Thread1BMS COLLEGE OF ENGINEERING
Thread2CSE
Thread2CSE
Thread2CSE
Thread2CSE
Thread2CSE
Thread1BMS COLLEGE OF ENGINEERING
CSExiting
Thread1BMS COLLEGE OF ENGINEERING
Thread1BMS COLLEGE OF ENGINEERING
Thread1BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERINGxiting
Main thread is awake
Main thread exiting
C:\Users\vedika\Desktop\javap>
```

Lab Program: 10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

WRITEUP

LAB PROG-10

Date _____
Page _____

Name - Vedika Dalmia, USN - IBM19CS181

```
import java.awt.*;
import java.awt.event.*;

public class Lab10 extends Frame implements
    ActionListener {
    TextField t1, t2;
    String msg = "";
    Button btn;

    Lab10() {
        Label L1 = new Label("First Number:",
            Label.RIGHT);
        t1 = new TextField(10);
        Label L2 = new Label("Second Number:", Label.RIGHT);
        t2 = new TextField(10);
        btn = new Button("Sum bit");

        L1.setBackground(Color.Blue);
        L2.setBackground(Color.Blue);

        this.add(L1);
        this.add(t1);
        this.add(L2);
        this.add(t2);
        this.add(btn, BorderLayout.CENTER);
        this.setVisible(true);
        this.setSize(600, 30);
        this.setLayout(new FlowLayout(FlowLayout.
            CENTER, 30, 20));

        btn.addActionListener(this);
        addWindowListener(new MyWindow());
        setBackground(Color.Blue);
    }
}
```

@ Override

```
public Insets getInsets ()
```

```
{
```

```
    return new Insets (50, 10, 10, 20);
```

```
}
```

```
public void actionPerformed (ActionEvent e)
```

```
{
```

```
    String st1 = t1.getText();
```

```
    String st2 = t2.getText();
```

```
    double n1, n2;
```

```
    n1 = 0.0;
```

```
    n2 = 0.0;
```

```
    if (st1.equals("") || st2.equals(""))
```

```
    {
```

```
        msg = "cannot leave text elements blank";
```

```
    }
```

```
    else
```

```
    {
```

```
        try {
```

```
            n1 = Double.parseDouble(st1);
```

```
            n2 = Double.parseDouble(st2);
```

```
        } catch {
```

```
            double res = n1/n2;
```

```
            msg = "Result of Division : " + res;
```

```
        }
```

```
        catch (ArithmeticException e1)
```

```
        {
```

```
            msg = e1.toString();
```

```
        }
```

```
    }
```

```
    catch (NumberFormatException e2)
```

```
    {
```

```
        msg = "Enter only numbers";
```

```
    }
```

```
}
```

Date _____
Page _____

```
new Dialog (this, "Result Dialog", false, msg, n1,  
            n2);
```

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

```
{  
    new Lab10();
```

```
}
```

```
}  
class MyDialog extends Dialog implements  
    ActionListener
```

```
{  
    public MyDialog (Frame owner, String title,  
        boolean modal, String msg, double n1, double n2)
```

```
{  
        super (owner, title, modal);  
        this.setVisible (true);  
        this.setSize (300, 400);  
        this.setLayout (new FlowLayout ());
```

```
        Label l1 = new Label ("Result updates :");
```

```
        this.add (l1);
```

```
        this.add (new Label ("First number : " + n1));
```

```
        this.add (new Label ("Second number : " + n2));
```

```
        this.add (new Label (msg));
```

```
        Button b = new Button ("Close");
```

```
        this.add (b);
```

```
        b.addActionListener (this);
```

```
        this.addWindowListener (new WindowAdapter())
```

```
{
```

```
        public void windowClosing (WindowEvent e)
```

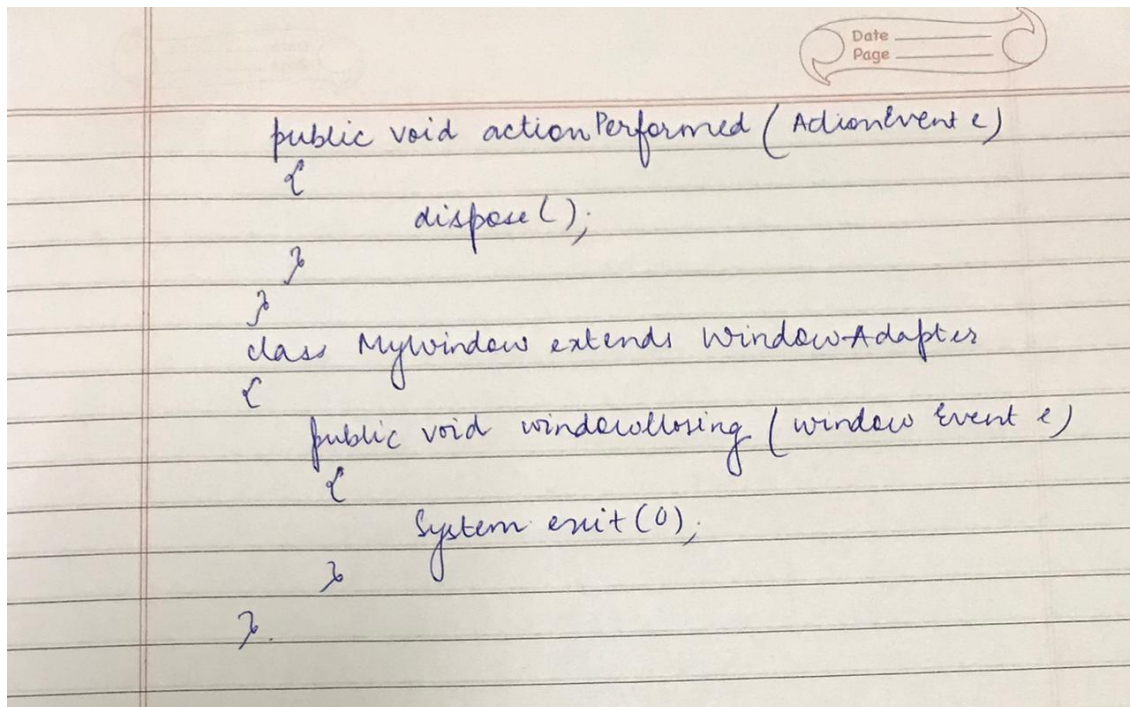
```
{
```

```
            dispose ();
```

```
}
```

```
}
```

```
}
```

PROGRAM

```

import java.awt.*;
import java.awt.event.*;
public class Lab10 extends Frame implements ActionListener{
    TextField t1,t2;
    String msg="";
    Button btn;
    Lab10(){
        Label l1 = new Label("First Number: ",Label.RIGHT);
        t1 = new TextField(10);
        Label l2 = new Label("Second Number: ",Label.RIGHT);
        t2 = new TextField(10);
        btn = new Button("Submit");
        //Label l = new Label("Updates:");
        l1.setBackground(Color.BLUE);
        l2.setBackground(Color.BLUE);
        //this.setResizable(false);
        this.add(l1);
        this.add(t1);
        this.add(l2);
    }
}

```

```
this.add(t2);
```

```
//the following command will make sure that the input char is not visible to the user
```

```
//(it has been added just to demonstrate). Can be used for passwords.
```

```
//t1.setEchoChar('*');
```

```
//t2.setEchoChar('#');
```

```
this.add(btn, BorderLayout.CENTER);
```

```
this.setVisible(true);
```

```
this.setSize(600, 300);
```

```
this.setLayout(new FlowLayout(FlowLayout.CENTER,20,10));
```

```
//t1.addActionListener(this);
```

```
btn.addActionListener(this);
```

```
addWindowListener(new MyWindow());
```

```
setBackground(Color.BLUE);
```

```
//System.out.println(BorderLayout.CENTER);
```

```
}
```

```
@Override
```

```
public Insets getInsets() {
```

```
return new Insets(50,10,10,20);
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
String st1 = t1.getText();
```

```
String st2 = t2.getText();
```

```
double n1,n2;
```

```

n1 = 0.0;
n2 = 0.0;
if(st1.equals("")||st2.equals("")) {

msg="You cannot leave the text elements blank";
}else{
try {
n1 = Double.parseDouble(st1);
n2 = Double.parseDouble(st2);
try {
double res = n1/n2;
msg = "Result of division: "+res;
}catch(ArithmeticException e1) {
msg = e1.toString();
}
}catch(NumberFormatException e2) {
msg = "Enter only numbers and not other things";
}
}
new MyDialog(this,"Result Dialog",false,msg,n1,n2);
}
public static void main(String[] args) {

new Lab10();
}
}

class MyDialog extends Dialog implements ActionListener{

public MyDialog(Frame owner, String title, boolean modal,String msg, double
n1,
double n2) {

super(owner, title, modal);
this.setVisible(true);
this.setSize(300, 400);
this.setLayout(new FlowLayout());

```

```
//System.out.println(owner);
Label l1 = new Label(" Updates on the result: ");
//l1.setSize(300, 20);
this.add(l1);
this.add(new Label("First Number: "+n1));
this.add(new Label("Second Number: "+n2));
this.add(new Label(msg));

Button b = new Button("Close");
this.add(b);
b.addActionListener(this);
this.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        dispose();
    }

});
}

@Override
public void actionPerformed(ActionEvent e) {
    dispose();

}

}

class MyWindow extends WindowAdapter{
    public void windowClosing(WindowEvent e) {

        System.exit(0);
    }
}
```

OUTPUT SCREEN:10

