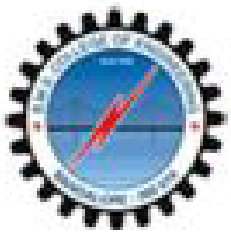
B.M.S. College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering

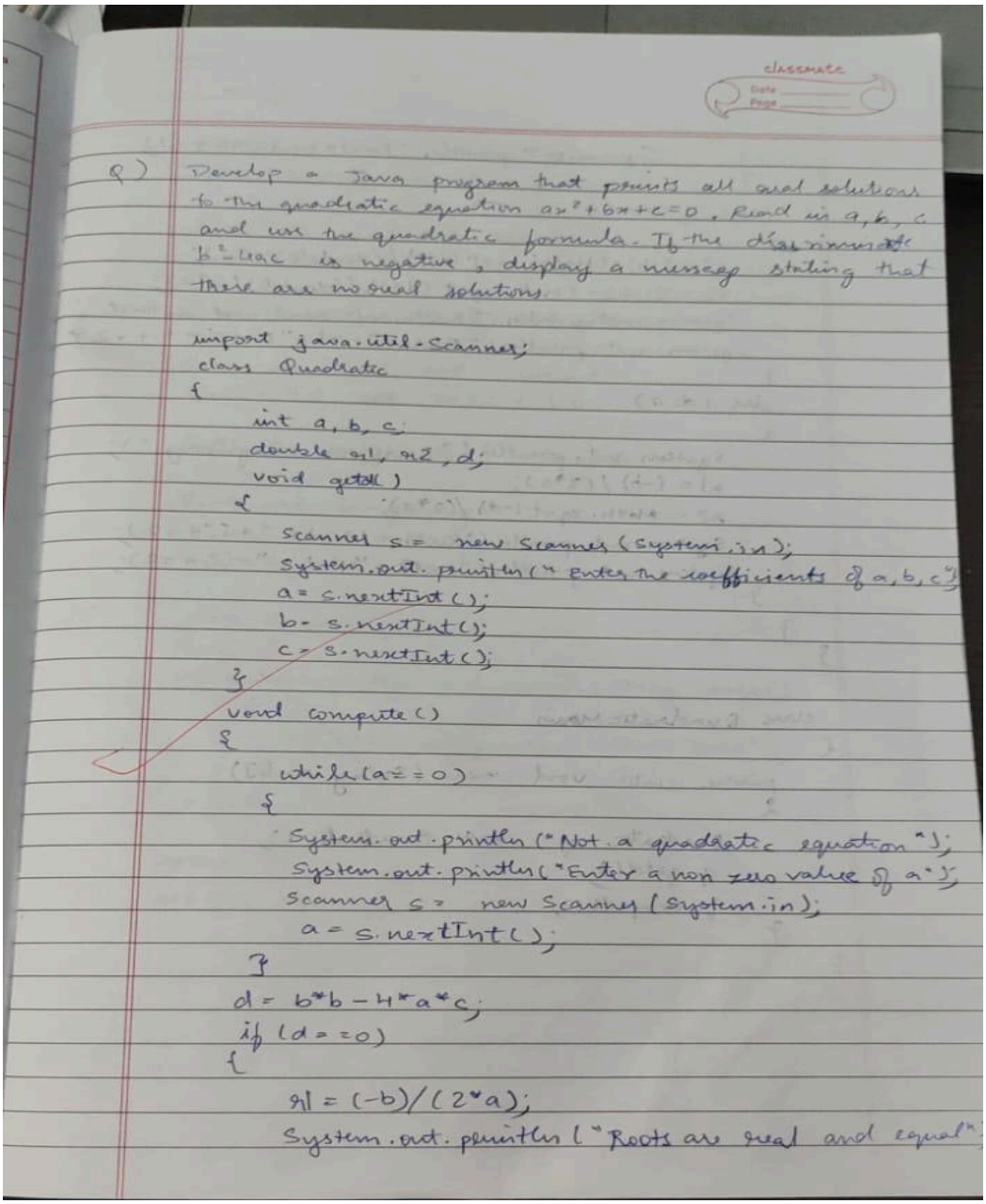


JAVA

Laboratory Experiments

NAME : MAHIKA D
CLASS : 3C
USN :1BM22CS142
BATCH : II

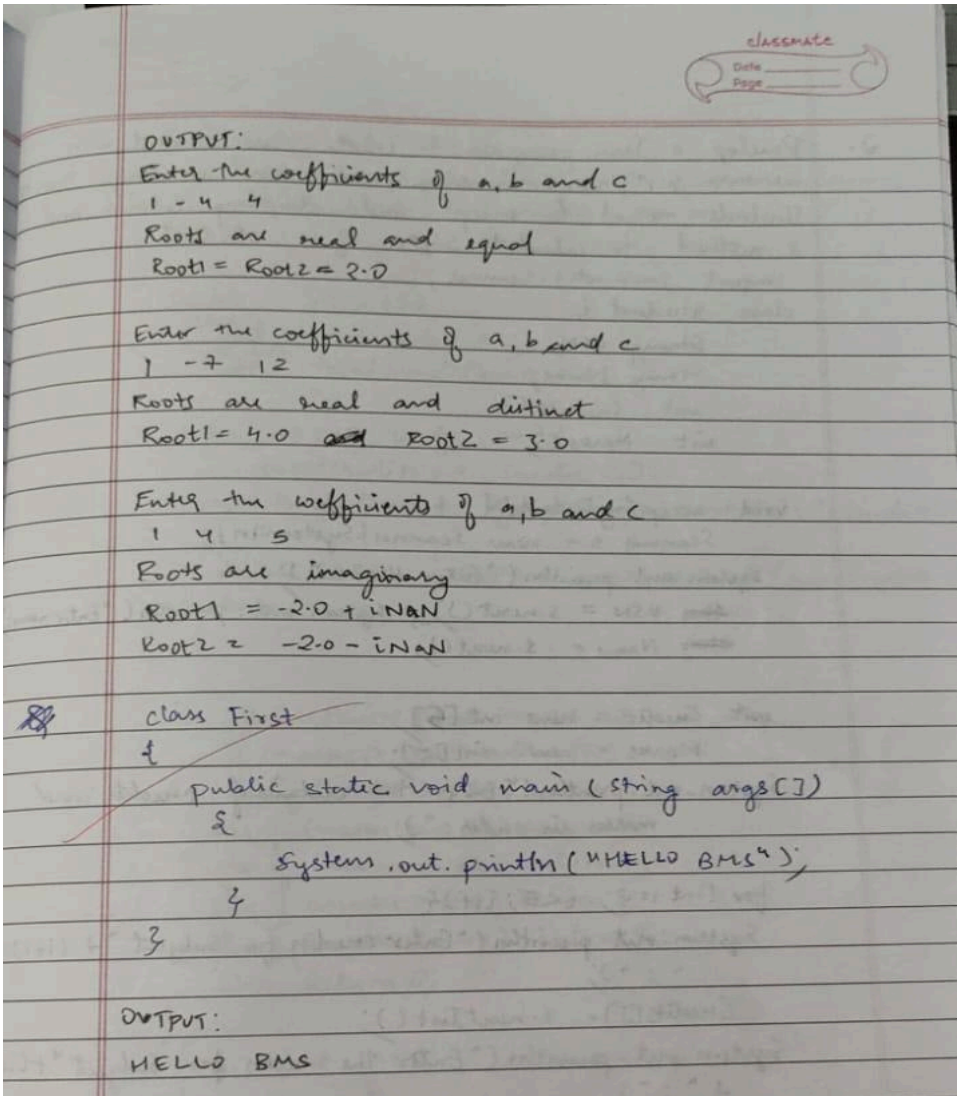
Q1) Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c
= 0. Read in a, b, c and use the quadratic formula. If the discriminant b2-4ac is negative,
display a message stating that there are no real solutions.



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

```
import java.util.Scanner;
class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getAll()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a, b, c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
        System.out.println("Not a quadratic equation");
        System.out.println("Enter a non zero value of a");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        }
    d = b*b - 4*a*c;
    if (d==0)
    {
        r1 = (-b)/(2*a);
        System.out.println("Roots are real and equal"
```

```
        System.out.println("Root1 = Root2 =" + r1);
    }
    else if (d>0)
    {
        r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
        r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
        System.out.println("Roots are real and distinct");
        System.out.println("Root1 =" + r1 + "Root2 =" + r2);
    }
    else (d<0)
    {
        System.out.println("Roots are imaginary");
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println("Root1 =" + r1 + "+i" + r2);
        System.out.println("Root2 =" + r1 + "-i" + r2);
    }
  }
}

class QuadraticMain
{
    public static void main (String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```



OUTPUT:
Enter the coefficients of a, b and c
1 -4 4
Roots are real and equal
Root1 = Root2 = 2.0

Enter the coefficients of a, b and c
1 -7 12
Roots are real and distinct
Root1 = 4.0 and Root2 = 3.0

Enter the coefficients of a, b and c
1 4 5
Roots are imaginary
Root1 = -2.0 + iNaN
Root2 = -2.0 - iNaN

class First
{
public static void main (String args[])
{
System.out.println("HELLO BMS");
}
}

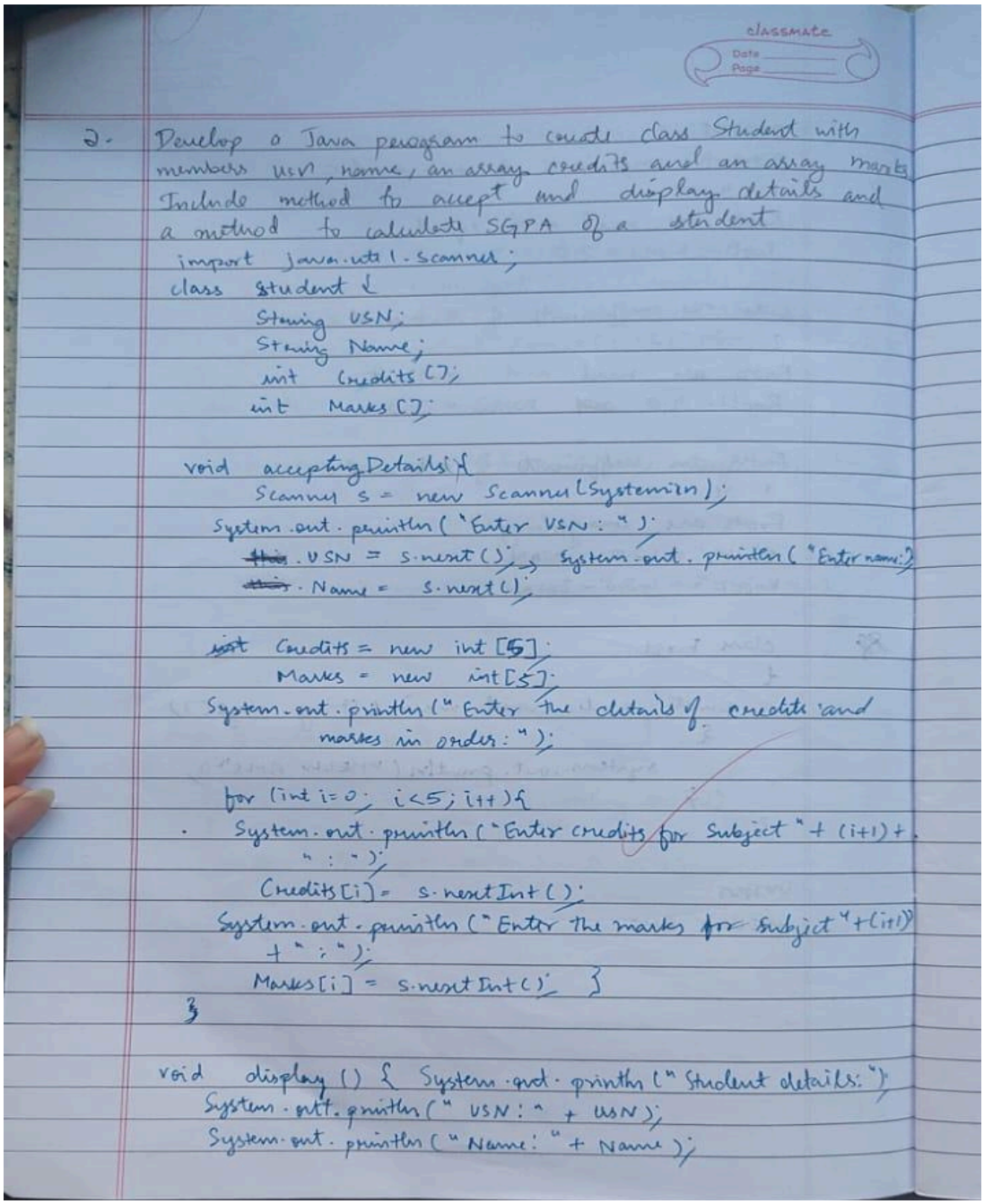
OUTPUT:
HELLO BMS

CODE :

```
import java.util.Scanner;
class Quadratic
{
int a,b,c;
double r1,r2,d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b and c");
a=s.nextInt();
b=s.nextInt();
c=s.nextInt();
```

```
}
void compute()
{
while(a==0)
{
System.out.println("Not a quadratic equation , enter a non zero value of a");
Scanner s= new Scanner(System.in);
a=s.nextInt();
}
d=b*b-4*a*c;
if(d==0)
{
r1=(-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Root1 = Root2 ="+r1);
}
else if(d>0)
{
r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root1 ="+r1+" Root2 ="+r2);
}
else
{
System.out.println("Roots are imaginary ");
r1=(-b)/(double)(2*a);
r2=(Math.sqrt(d))/(double)(2*a);
System.out.println("Root1 ="+r1+ "+ i"+r2);
System.out.println("Root2 ="+r1+ "- i"+r2);
}
}
}
class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
}
}
```

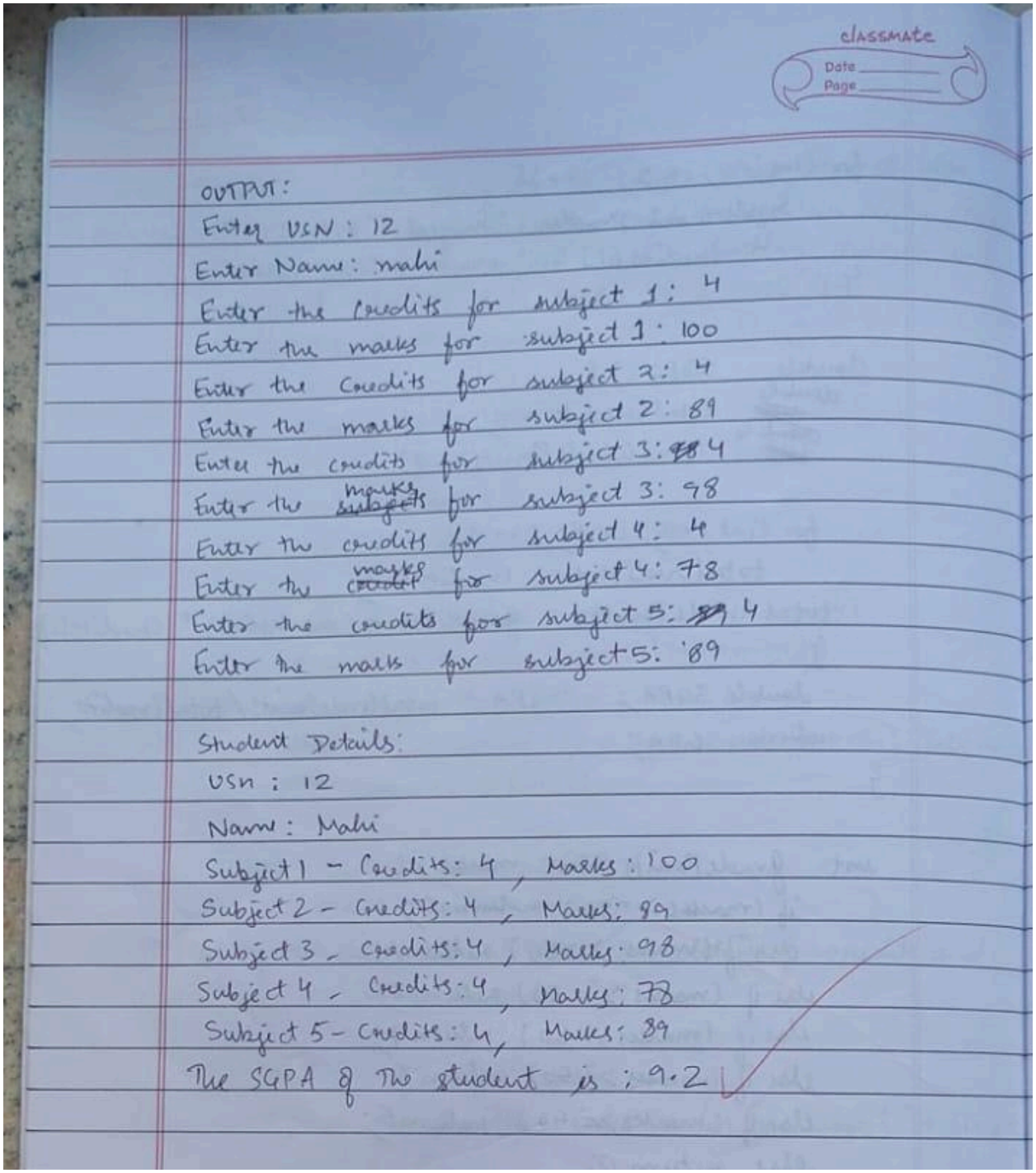
Q2) Develop a Java program to create a class Student with members usn, name, an array
credits and an array marks. Include methods to accept and display details and a
method to calculate SGPA of a student.



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

```
import java.util.scanner;
class student {
    String USN;
    String Name;
    int Credits[];
    int Marks[];

void acceptingDetails(){
    Scanner s = new Scanner(System.in);
System.out.println("Enter USN: ");
    this.USN = s.next();  System.out.println("Enter name:");
    this.Name = s.next();

int Credits = new int[5];
    Marks = new int[5];
System.out.println("Enter the details of credits and
        marks in order:");

for(int i=0; i<5; i++){
  System.out.println("Enter credits for Subject "+(i+1)+
      ":");
    Credits[i] = s.nextInt();
System.out.println("Enter the marks for Subject "+(i+1)
    +":");
    Marks[i] = s.nextInt(); }
}

void display(){ System.out.println("Student details:");
  System.out.println("USN: "+USN);
  System.out.println("Name: "+Name);
```

```
for (int i=0; i<5; i++){
    System.out.println("Subject: " + (i+1) + " - Credits: "
    + Credits[i] + " Marks: " + Marks[i]); }
}

double SGPA(){
    double totalCredits = 0;
    double totalGradePoints = 0;

    for (int i=0; i<5; i++){
        totalCredits += Credits[i];
    totalGradePoints += gradePoints(marks[i]) * Credits[i];
    }
    double SGPA; SGPA = totalGradePoints/totalCredits;
    return SGPA;
}

int GradePoints (int marks){
    if (marks >= 90) return 10;
    else if (marks >= 80) return 9;
    else if (marks >= 70) return 8;
    else if (marks >= 60) return 7;
    else if (marks >= 50) return 6;
    else if (marks >= 40) return 5;
    else return 0;
}

class Main {
public static void main (String args[]){
    Student object = new Student();
    object.acceptingDetails();
    object.display();
    System.out.println("SGPA: " + object.SGPA());
}
}
```



OUTPUT:

Enter USN : 12
Enter Name : mahi
Enter the credits for subject 1 : 4
Enter the marks for subject 1 : 100
Enter the credits for subject 2 : 4
Enter the marks for subject 2 : 89
Enter the credits for subject 3 : 4
Enter the marks for subject 3 : 98
Enter the credits for subject 4 : 4
Enter the marks for subject 4 : 78
Enter the credits for subject 5 : 4
Enter the marks for subject 5 : 89

Student Details:
USN : 12
Name : Mahi
Subject 1 - Credits : 4 , Marks : 100
Subject 2 - Credits : 4 , Marks : 89
Subject 3 - Credits : 4 , Marks : 98
Subject 4 - Credits : 4 , Marks : 78
Subject 5 - Credits : 4 , Marks : 89
The SGPA of the student is : 9.2

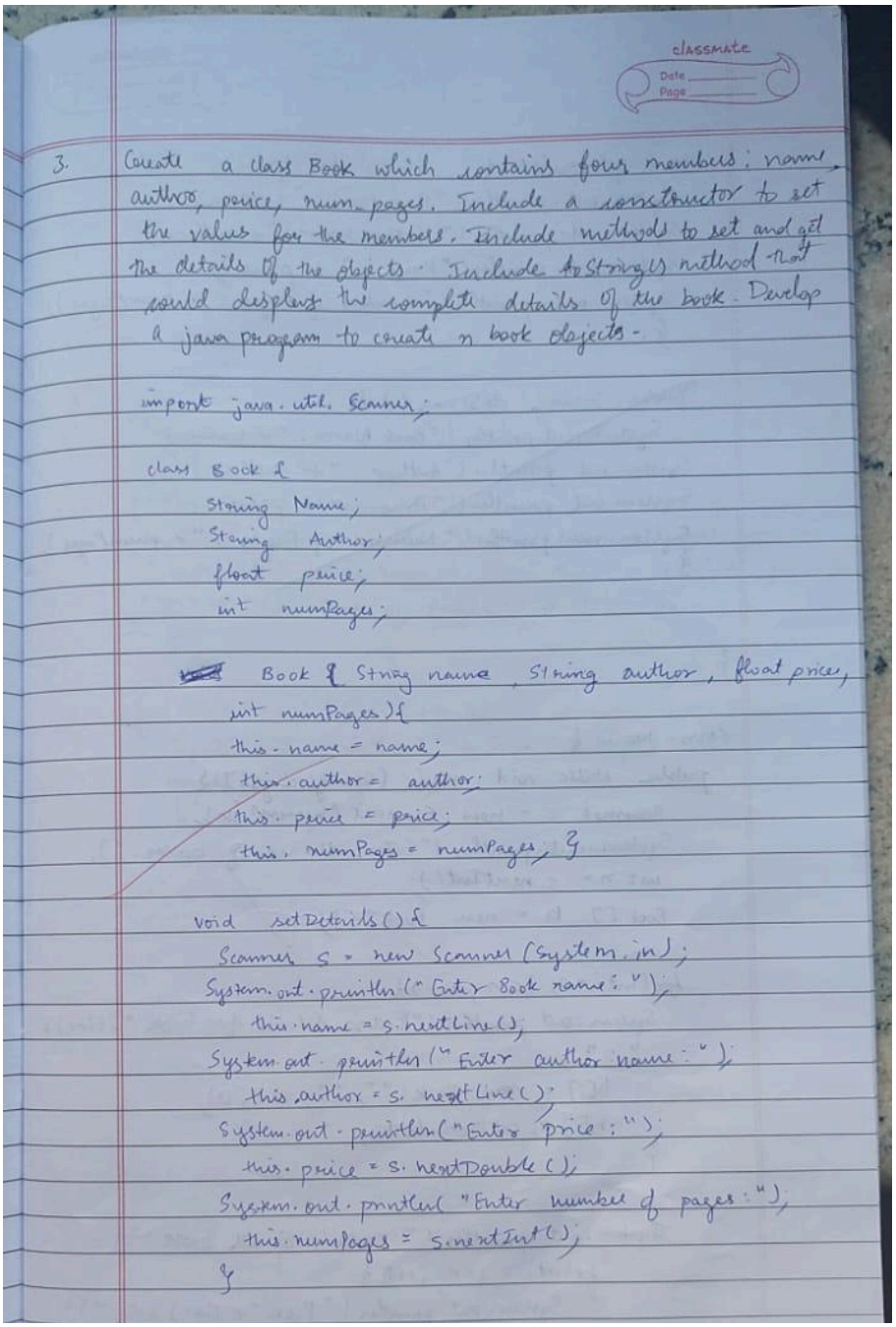
CODE :

```
import java.util.Scanner;
class Student {
    String usn,name;
    int[] credits,marks;
```

```
void acceptDetails() {
    Scanner hello = new Scanner(System.in);

    System.out.print("Enter USN: ");
    this.usn = hello.next();

    System.out.print("Enter Name: ");
    this.name = hello.next();

    credits = new int[4];
    marks = new int[4];

    System.out.println("Enter details of credits and marks in order for 4 subjects:");

    for (int i = 0; i < 4; i++) {
        System.out.print("Enter credits for Subject " + (i + 1) + ": ");
        credits[i] = hello.nextInt();

        System.out.print("Enter marks for Subject " + (i + 1) + ": ");
        marks[i] = hello.nextInt();
    }
}

void display() {
    System.out.println("Student Details.");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);

    for (int i = 0; i < 4; i++) {
        System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);
    }
}


double calc() {
    int totalCredits = 0;
    double totalGradePoints = 0;

    for (int i = 0; i < 4; i++) {
        totalCredits += credits[i];
        totalGradePoints += gradePoints(marks[i]) * credits[i];
    }
```

```
        return totalGradePoints / totalCredits;
    }

    int gradePoints(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
}

public class SGPA {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        System.out.println("\nStudent Details:");
        student.display();
        System.out.println("\nSGPA: " + student.calc());
    }
}
```

Q3) 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.



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.util.Scanner;

class Book {
    String Name;
    String Author;
    float price;
    int numPages;

    Book { String name, String author, float price,
    int numPages ){
    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages; }

    void setDetails() {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter Book name: ");
        this.name = s.nextLine();
    System.out.println("Enter author name: ");
        this.author = s.nextLine();
    System.out.println("Enter price: ");
        this.price = s.nextDouble();
    System.out.println("Enter number of pages: ");
        this.numPages = s.nextInt();
    }
```

```
void getDetails(){
System.out.println("Book Name: " + name);
System.out.println("Author Name: " + author);
System.out.println("Price: " + price);
System.out.println("Number of Pages: " + numPages);
}

public String toString() {
System.out.println("Book Name: " + name);
System.out.println("Author: " + author);
System.out.println("Price: " + price);
System.out.println("Number of Pages: " + numPages);
}

}

class Main {
public static void main(String args[]){
Scanner s = new Scanner(System.in);
System.out.println("Enter the no. of books: ");
int n = s.nextInt();
Book[] b = new Book[n];

for(int i=0; i<n; i++){
System.out.println("Enter details for book " + (i+1) +
": ");
b[i] = new Book("", "", 0.0, 0);
b[i].setDetails();
}

System.out.println("Details of all books:");
for(int i=0; i<n; i++) {
System.out.println("Book " + (i+1) + ":");
```

```
        b[i].getDetails();
    }
}
```

```
public String toString(){
    return "Book Details:\n" + "Name: " + name +
    "Author: " + author + "\n" + "Price: " + price +
    "\n" + "Number of Pages: " + numPages;
    }
```

```
for (int i=0; i<n; i++){
    System.out.println("\n Book "+(i+1)+ ":\n" +
    books[i].toString());
    }
```

OUTPUT:

```
Enter the number of books : 2

Enter details for Book 1 :
Enter book name : M
Enter author name : A
Enter price : 12
Enter number of pages: 34

Enter details for Book 2 :
Enter book name : H
Enter author name : I
Enter price : 34
Enter number of pages : 46

Details of all books :

Book 1 :
Book Name : M
Author Name : A
Price : 12.0
Number of pages : 34

Book 2 :
Book Name : H
Author Name : I
Price : 34.0
Number of Pages : 46
```

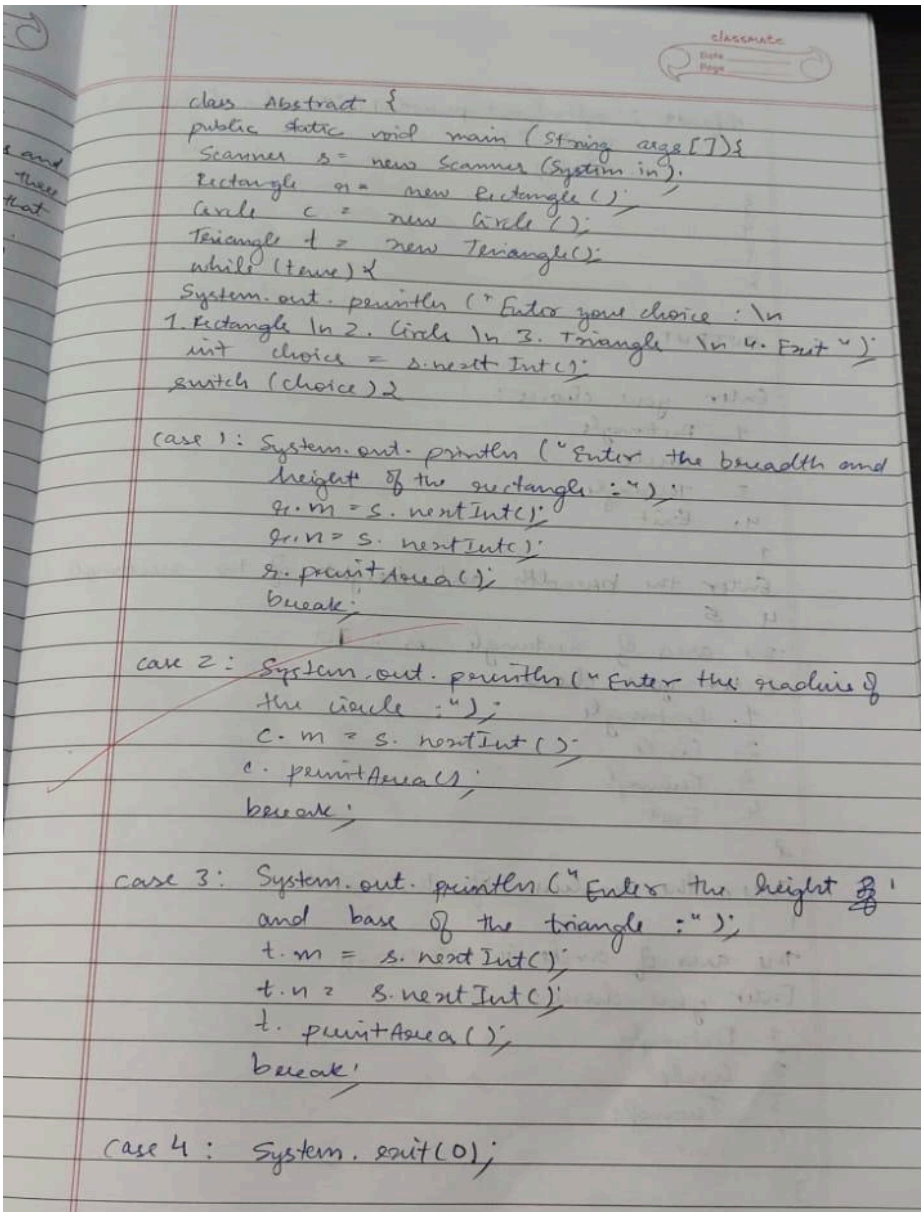
```
Book1:
Book details using toString method:
Name : M
Author: A
Price: 12.0
Number of Pages: 34

Book 2:
Book details using toString method:
Name: H
Author: I
Price: 34.0
Number of Pages: 46
```

CODE :

```
import java.util.Scanner;

class Book {
String name;
String author;
double price;
int numPages;

public Book(String name, String author, double price, int numPages) {
this.name = name;
this.author = author;
this.price = price;
this.numPages = numPages;
    }


    public void setDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter book name: ");
        this.name = scanner.nextLine();

        System.out.print("Enter author name: ");
        this.author = scanner.nextLine();

        System.out.print("Enter price: ");
        this.price = scanner.nextDouble();

        System.out.print("Enter number of pages: ");
        this.numPages = scanner.nextInt();
    }

    public void getDetails() {
        System.out.println("Book Name: " + name);
        System.out.println("Author: " + author);
        System.out.println("Price: $" + price);
        System.out.println("Number of Pages: " + numPages);
    }


    public String toString() {
```

```
        return "Book Details using toString method:\n" +"Name: " + name + "\n" +"Author: " +
author + "\n" +"Price: " + price + "\n" +"Number of Pages: " + numPages    ;
    }
}

class BOOK1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();


        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            books[i] = new Book("", "", 0.0, 0);
            books[i].setDetails();
        }


        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":");
            books[i].getDetails();
        }

        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
        }
    }
}
```

Q4) 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.

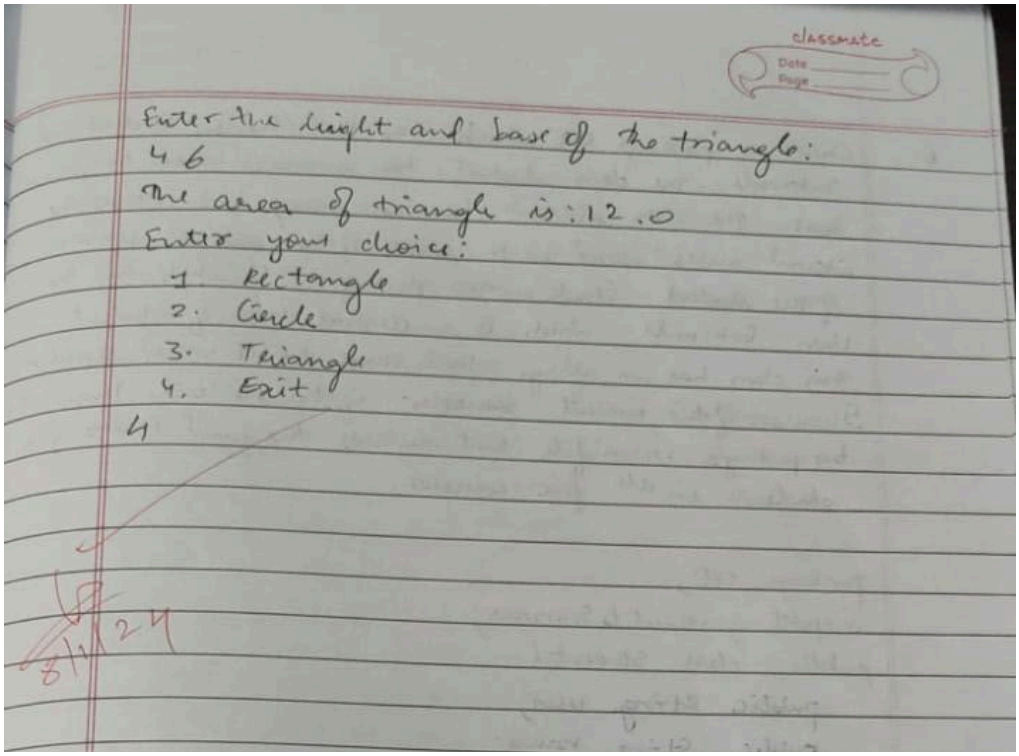


```
class Abstract {
public static void main (String args[]){
Scanner s= new Scanner (System.in);
Rectangle r= new Rectangle ();
Circle c = new Circle ();
Triangle t = new Triangle();
while (true){
System.out.println ("Enter your choice : \n
1. Rectangle \n 2. Circle \n 3. Triangle \n 4. Exit ");
int choice = s.nextInt();
switch (choice){

case 1: System.out.println ("Enter the breadth and
        height of the rectangle :");
        r.m = s.nextInt();
        r.n = s.nextInt();
        r.printArea();
        break;

case 2: System.out.println ("Enter the radius of
        the circle :");
        c.m = s.nextInt();
        c.printArea();
        break;

case 3: System.out.println ("Enter the height
        and base of the triangle :");
        t.m = s.nextInt();
        t.n = s.nextInt();
        t.printArea();
        break;

case 4: System.exit(0);
```

```
default : System.out.println("Invalid choice. Enter
          again");
          break;
}
}
}
}
```

OUTPUT:

```
Enter your choice:
1. Rectangle
2. Circle
3. Triangle
4. Exit
1
Enter the breadth and height of the rectangle:
4 5
The area of rectangle is : 20
Enter your choice:
1. Rectangle
2. Circle
3. Triangle
4. Exit
2
Enter the radius of the circle:
1
The area of circle is : 3.14285
Enter your choice:
1. Rectangle
2. Circle
3. Triangle
4. Exit
3
```



CODE :

```
import java.util.Scanner;
abstract class Shape{
int m , n ;
abstract void printArea();
}

class Rectangle extends Shape{
void printArea(){
System.out.println("The area of rectangle is :"+(m*n));
}
}

class Circle extends Shape{
void printArea(){
System.out.println("The area of circle is :"+((22.0/7)*m*m));
}
}


class Triangle extends Shape{
void printArea(){
```

```
System.out.println("The area of triangle is :"+((1/2.0)*m*n));
}
}

class Abstract{
public static void main(String args[]){
Scanner s = new Scanner(System.in);
Rectangle r = new Rectangle();
Circle c = new Circle();
Triangle t=new Triangle();
while(true){
System.out.println("Enter your choice : \n 1.Rectangle \n 2.Circle \n 3.Triangle \n 4.Exit ");
int choice = s.nextInt();
switch(choice)
{
case 1 : System.out.println("Enter The breadth and height of the rectangle :");
          r.m=s.nextInt();
          r.n=s.nextInt();
          r.printArea();
          break;

case 2 : System.out.println("Enter the radius of the circle :");
          c.m=s.nextInt();
          c.printArea();
          break;

case 3 : System.out.println("Enter The height and base of the triangle :");
          t.m=s.nextInt();
          t.n=s.nextInt();
          t.printArea();
          break;

case 4 :System.exit(0);

default :System.out.println("Invalid choice . Enter again");
          break;
}
}
}
}
```

Q5) 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

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

19/02/2024

Q5.

```
import java.util.Scanner;
class Bank
{ float balance;}
class Account extends Bank
{ String cus_name, acc_type; int acc_no;
  Account (String name, int acc_no, String acc_type)
  { this.cus_name = name;
    this.acc_no = acc_no;
    this.acc_type = acc_type; balance = 0; }}

void setBal (float amt)
 { balance += amt; }
void disBal (){
 System.out.println ("Balance : Rs" + this.balance);}

class sav_acct extends Account
{ float comp_int, withdrawal;
  sav_acct (String name, int acc_no, String acc_type)
 { super (name, acc_no, acc_type); }
  void inter (float rate)
 { System.out.println ("Interest :" + (balance* rate /100);
    balance += (balance * rate /100); }}

class cu_acct extends Account
 { float comp_int, withdrawal;
  cu_acct (String name, int acc_no, String acc_type)
 { super (name, acc_no, acc_type); balance = 0.0f;}
 void with (float amt ){
    if (amt < balance)
     balance -= amt;
   else
     System.out.println ("Amount exceeds balan
 if (balance <500)
 { System.out.println ("No minimum balance
```

```
maintained ! Rs 500 fine levied ");
    balance -=500;  }}
```

```
class Lab05
{ public static void main( String args[]) {
  Scanner s = new Scanner (System.in);
  int k = 0; j=0; int ch; boolean t = true;
  System.out.println ("Enter the no. of customers :");
  int n= s.nextInt();
  sav_acct a[] = new sav_acct [n];
  cur_acct b[] = new cur_acct [n];
  for (int i= 0; i<n; i++)
  { System.out.println ("Enter the name, Account
  type and Account Number :");
  String name = s.nextLine();
  String acc_type = s.nextLine();
  int acc_no = s.nextInt();
    if (acc_type.equalsIgnoreCase ("savings"))
  { a[k] = new sav_acct (name, acc_no, acctype);
    while (t)
  { System.out.println ("Enter 1. Update Balance \n 2.
  Display Balance \n 3. Compute and deposit interest
  \n 4. End ");
      ch = s.nextInt();
    switch (ch) {
  case 1: System.out.println ("Enter the deposit amount
      a[k].setBal (s.nextFloat()); break;
  case 2: a[k].disBal(); break;
  case 3: System.out.println ("Enter the rate of interest :"
      a[k].intcq (s.nextFloat()); break;
  case 4: t = false; break;
  default: System.out.println ("Invalid choice") }
    k++; } t = true;
```

```
if(acc_type.equalsIgnoreCase("current"))
{ b[j]=new cur_acct(name, acc_no, acc_type);
  while(t)
{ System.out.println("Enter 1. Update balance \n 2.
Display Balance \n 3. Withdraw and Update balance
 \n 4. End ");
    ch=s.nextInt();
    switch(ch){
case 1 : System.out.println("Enter the deposit amount:");
    b[j].setBal(s.nextFloat()); break;
case 2 : b[j].dispBal(); break;
Case 3: System.out.println("Enter the withdraw amount");
    b[j].with(s.nextFloat()); break;
case 4: t=false; break;
default : System.out.println("Invalid choice");
}} j++; } } time; } } }
```

OUTPUT:

Enter the number of customers: 2

Enter the name, Account type and Account number of A

Savings

123

Enter 1. Update Balance

2. Display Balance.

3. Compute and deposit interest

4. End

1

Enter the deposit amount;

1200

Enter 1. Update balance

2. Display balance;

3. Compute and deposit interest

4. End
3
Enter the rate of interest :
8
Interest : 96.0
Enter 1. Update balance
2. Display balance
3. Compute and deposit interest
4. End
2
Balance : Rs 1296.0
Enter 1. Update balance
2. Display balance
3. Compute and deposit interset
4. End
4
Enter the name, Account type and Account number of user2;
B
current
234
Enter 1. Update balance
2. Display balance
3. Withdraw and update balance
4. End
1
Enter the deposit amount :
1200
Enter 1. Update balance
2. Display balance
3. withdraw and update balance
4. End
2
Balance : Rs1200.0

Enter 1. Update Balance
2. Display Balance
3. Withdraw & update Balance
4. End.
3
Enter the withdraw amount:
500
Enter 1. Update balance
2. Display balance
3. Withdraw & update Balance
4. End
2
Balance : Rs 700.0

Enter 1.
4.
3
Enter the withdraw amount:
701
Amount exceeds balance!

Enter 1.
4.
3
Enter the withdraw amount:
600
No minimum balance maintained ! Rs 500 fine levied
Enter 1.
4.
2
Balance : Rs -400.0

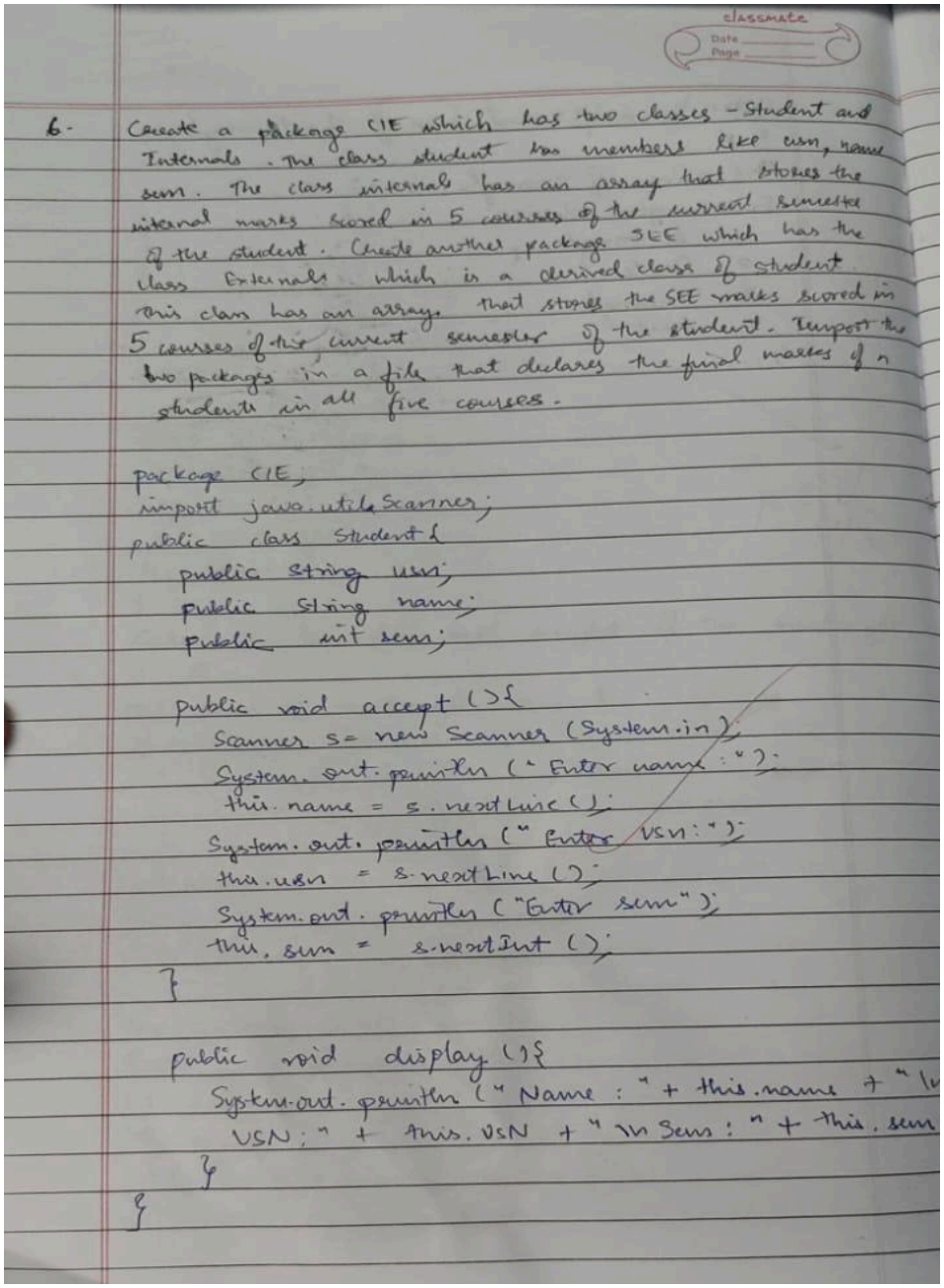
CODE:

```
import java.util.Scanner;
class Bank
{
   float balance;
}

class Account extends Bank
{
   String cus_name,acc_type;int acc_no;
   Account(String name,int acc_no,String acc_type)
   {
      this.cus_name=name;this.acc_no=acc_no;this.acc_type=acc_type;balance=0.0f;
   }
   void setBal(float amt)
   {
      balance+=amt;
   }
   void disBal(){
   System.out.println("Balance:Rs "+this.balance);
   }
}

class Sav_acct extends Account
{
   float comp_int,withdrawal;
   Sav_acct(String name,int acc_no,String acc_type)
   {
      super(name,acc_no,acc_type);
   }
   void inter(float rate)
   {
      System.out.println("Interest:"+(balance*rate/100));
      balance+=(balance*rate/100);
   }
}

class Cur_acct extends Account
{
   float comp_int,withdrawal;
   Cur_acct(String name,int acc_no,String acc_type)
   {
```

```
        super(name,acc_no,acc_type);
            balance=0.0f;
    }
    void with(float amt){
        if(amt<balance)
        balance-=amt;
        else
        System.out.println("Amount exceeds balance!");
    if(balance<500)
        {
            System.out.println("No minimum balance maintained!Rs.500 fine levied.");
            balance-=500;
        }
    }
}

class LabQ5
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int k=0,j=0;int ch;boolean t=true,t1=true;
        System.out.println("Enter the number of customers:");
        int n=sc.nextInt();
        Sav_acct a[]=new Sav_acct[n];Cur_acct b[]=new Cur_acct[n];

        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the name, Account type and Account number of user"
+(i+1)+":");
            String name=sc.next();
            String acc_type=sc.next();
            int acc_no=sc.nextInt();

            if(acc_type.equalsIgnoreCase("savings"))
            {
                a[k]=new Sav_acct(name,acc_no,acc_type);
                while(t)
                {
                    System.out.println("Enter 1.Update balance.\n2.Display balance.\n3.Compute and
deposit interest.\n4.End");
                    ch=sc.nextInt();
                    switch(ch){
                    case 1:System.out. println("Enter the deposit amount:");
```

```
                a[k].setBal(sc.nextFloat());
                break;
                case 2:
                a[k].disBal();
                break;
                case 3:System.out.println("Enter the rate of interest:");
                a[k].inter(sc.nextFloat());
                break;
                case 4:t=false;break;
                default:System.out.println("Invalid choice");
                }
            }
            k++;
        }

        if(acc_type.equalsIgnoreCase("current"))
        {
            b[j]=new Cur_acct(name,acc_no,acc_type);
            while(t1)
            {
                System.out.println("Enter 1.Update balance.\n2.Display balance.\n3.withdraw and
update balance.\n 4.End");
                ch=sc.nextInt();
                switch(ch){
                case 1:System.out. println("Enter the deposit amount:");
                b[j].setBal(sc.nextFloat());
                break;
                case 2:
                b[j].disBal();
                break;
                case 3:System.out. println("Enter the withdraw amount:");
                b[j].with(sc.nextFloat());
                break;
                case 4:t1=false;break;
                default:System.out.println("Invalid choice");
                }
            }j++;
        }
    }
  }
}
```

Q6)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.



6- Create a package CIE which has two classes - Student and Internals. The class student has members like usn, name sem. The class internals has an array that stores the internal marks scored in 5 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 5 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.

```
package CIE;
import java.util.Scanner;
public class Student{
    public String usn;
    public String name;
    public int sem;

    public void accept(){
        Scanner s= new Scanner(System.in);
        System.out.println("Enter name:");
        this.name = s.nextLine();
        System.out.println("Enter USN:");
        this.usn = s.nextLine();
        System.out.println("Enter sem");
        this.sem = s.nextInt();
    }

    public void display(){
        System.out.println("Name : " + this.name + "\n
        USN: " + this.USN + "\n Sem : " + this.sem
    }
}
```

```
package CIE;
import java.util.Scanner;
public class Internal extends CIE.Student {
    public int arr[] = new int[5];
    CIE.Student Student = new CIE.Student();
    public void accept() {
        Student.Accept();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal marks:");
        for(int i=0; i<5; i++){
            arr[i] = s.nextInt();
    }
}

public void display(){
    Student.display();
    for(int i=0; i<5; i++){
        System.out.println("Marks of subject "+(i+1)+
        " = "+arr[i]);
    }
}
}

package SEE;
import java.util.Scanner;
import CIE.Student;

public class External extends CIE.Student {
    public int x[] = new int[5];
    public void accept(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External marks:");
        for(int i=0; i<5; i++){
            x[i] = s.nextInt();
}
}
```

```
public void display (){
    super.display();

    for(int i=0; i<5; i++){
        System.out.println("Marks of sub" + (i+1) + " = "
            m[i]);
    }
}
}
```

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

import java.util.Scanner;
import CIE.*;
import SEE;

public class Final {
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter n:");
        int n = s.nextInt();

CIE.Internal[] c1 = new CIE.Internal[n];
SEE.External[] c2 = new SEE.External[n];

for(int i=0; i<n; i++){
    c1[i] = new CIE.Internal();
    c2[i] = new SEE.External();

    c1[i].accept();
    c2[i].accept();
    c1[i].display();
    c2[i].display();
```

```
for(int j=0 ; j<5; j++){
    double calc = c1[i].marks[j] + ((c2.[i].x[j])/2);
    System.out.println("Final marks of subject "+(j+1)+
    " = "+ calc);
}
}
}
```

OUTPUT:

Enter Name: aish
Enter usn: 123
Enter sem: 3
Enter Internal Marks:
10 20 30 40 50
Enter External Marks:
100 90 50 70 60

Final marks of sub1 = 60.0
Final marks of sub2 = 65.0
Final marks of sub3 = 70.0
Final marks of sub4 = 75.0
Final marks of sub5 = 80.0

Details:
Name : aish
USN : 123
Sem : 3
Marks of sub1 = 10
Marks of sub2 = 20
Marks of sub3 = 30
Marks of sub4 = 40
Marks of sub5 = 50

External Marks:
Marks of sub1 = 100
Marks of sub2 = 90
Marks of sub3 = 80
Marks of sub4 = 70
Marks of sub5 = 60

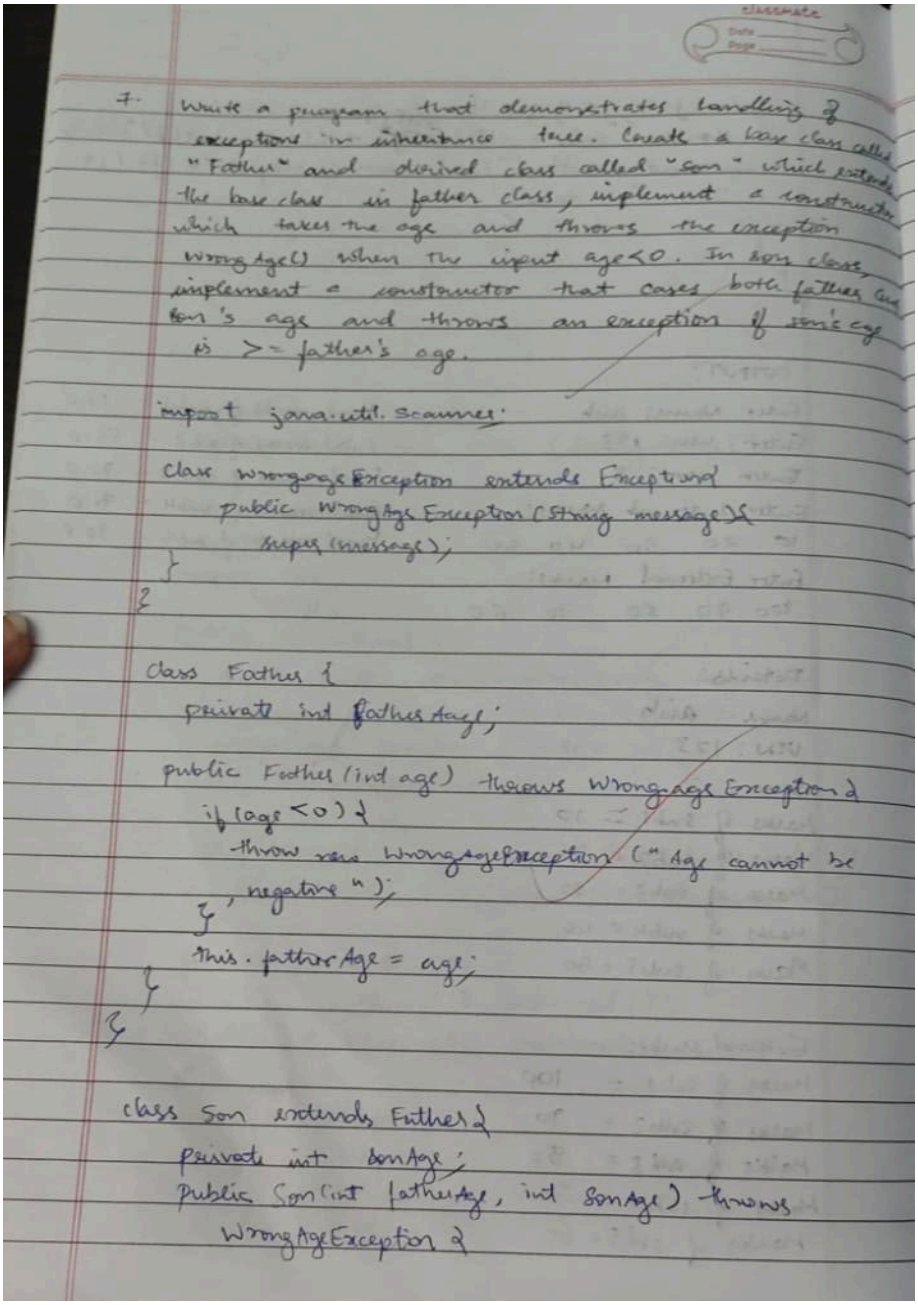
CODE:

```
package CIE;

import java.util.Scanner;

public class Internal extends CIE.Student{
   public int m[] = new int[5];
   CIE.Student student = new CIE.Student();
   public void accept(){
      student.accept();

      Scanner s1 = new Scanner(System.in);
      System.out.println("Enter Internal Marks:");
      for(int i=0;i<5;i++){
         m[i] = s1.nextInt();
      }
   }

   public void display(){

      student.display();
      for(int i=0;i<5;i++){
         System.out.println("Marks of sub" + (i+1) + " = " + m[i]);
      }
   }
}


package SEE;
import java.util.Scanner;

import CIE.Internal;
import CIE.Student;



public class External extends CIE.Student{
   public int x[] = new int[5];
   public void accept(){
      Scanner s2 = new Scanner(System.in);
      System.out.println("Enter External Marks:");
      for(int i=0;i<5;i++){
         x[i] = s2.nextInt();
```

```
        }
    }

    public void display(){
        for(int i=0;i<5;i++){
            System.out.println("Marks of sub" + (i+1) + " = " + x[i]);
        }
    }

}


package CIE;
import java.util.Scanner;

public class Student{
    public String usn;
    public String name;
    public int sem;


    public void accept(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        this.name = s.nextLine();
        System.out.println("Enter usn:");
        this.usn = s.nextLine();
        System.out.println("Enter sem");
        this.sem = s.nextInt();
    }

    public void display(){
        System.out.println("Name: " + this.name + "\nUSN: " + this.usn + "\nSem: " + this.sem);
    }
}


import java.util.Scanner;

import CIE.Student;
import CIE.Internal;
import SEE.External;

public class Final{
```

```
    public static void main(String[] args) {
        Scanner n = new Scanner(System.in);

        System.out.println("Enter n:");
        int y = n.nextInt();

        CIE.Internal[] c1 = new CIE.Internal[y];
        SEE.External[] c2 = new SEE.External[y];

        for(int i=0;i<y;i++){
            c1[i] = new CIE.Internal();
            c2[i] = new SEE.External();

            c1[i].accept();
            c2[i].accept();

            // c1[i].accept();c2[i].accept();
            c1[i].display();c2[i].display();

            for(int j=0;j<5;j++){
                double calc = c1[i].m[j]+((c2[i].x[j])/2);
                System.out.println("Final marks of sub["+(j+1)+"]= "+calc);
            }
        }
    }
}
```

Q7)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.



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

```
import java.util.Scanner;

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

class Father {
    private int fatherAge;

    public Father(int age) throws WrongAgeException {
        if(age<0){
            throw new WrongAgeException("Age cannot be negative");
        }
        this.fatherAge = age;
    }
}

class Son extends Father{
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws
        WrongAgeException {
```

```
super (fatherAge);
if (sonAge >= fatherAge){
  throws new WrongAgeException ("Son's age should
  be less than Father's age");
}
this.sonAge = sonAge;
System.out.println ("Father's Age : " + fatherAge);
System.out.println ("Son's Age : " + sonAge);

public class ExceptionInheritance {
  public static void main (String args[]) {
  Scanner s = new Scanner (System.in);
  try {
    System.out.println ("Enter Father's Age : ");
    int fatherAge = s.nextInt();
    Father father = new Father (fatherAge);

    System.out.print ("Enter Son's Age : ");
    int sonAge = Scanner.nextInt ();
    Son son = new Son (fatherAge, sonAge);
  }
  catch (WrongAgeException e) {
    System.out.println ("Exception : " + e.getMessage
    ());
  }
}
```

OUTPUT:

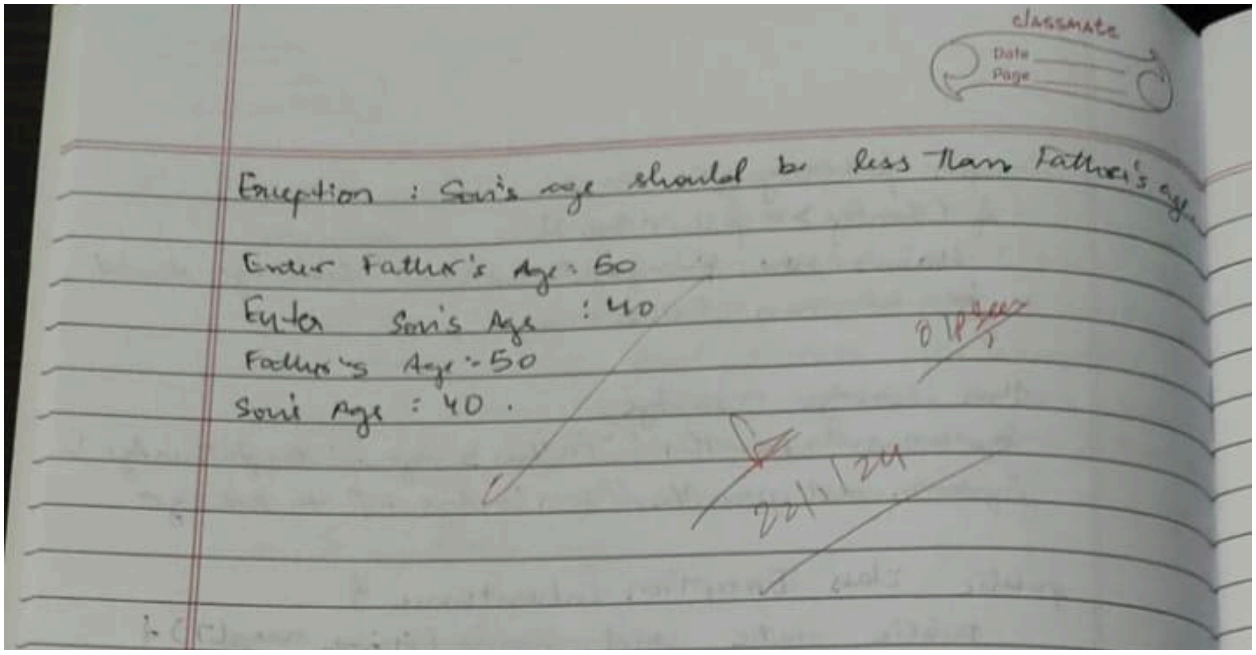
Enter Father's age = -1

Exception : Age cannot be negative

Enter Father's Age : 50

~~Father's Ag~~

Enter Son's Age : 59

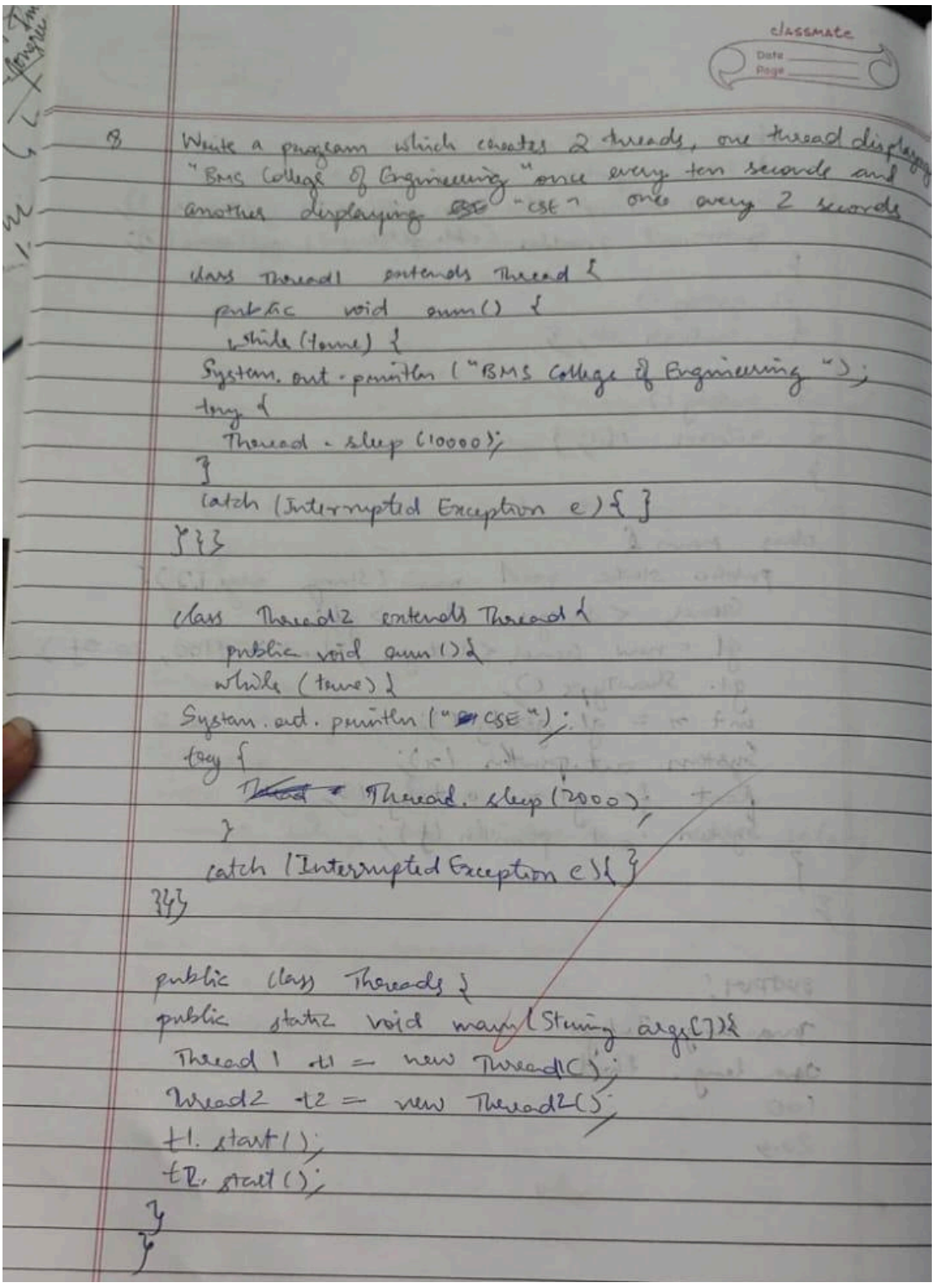


CODE :

```
import java.util.Scanner;

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

class Father {
    private int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        this.fatherAge = age;

    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
```

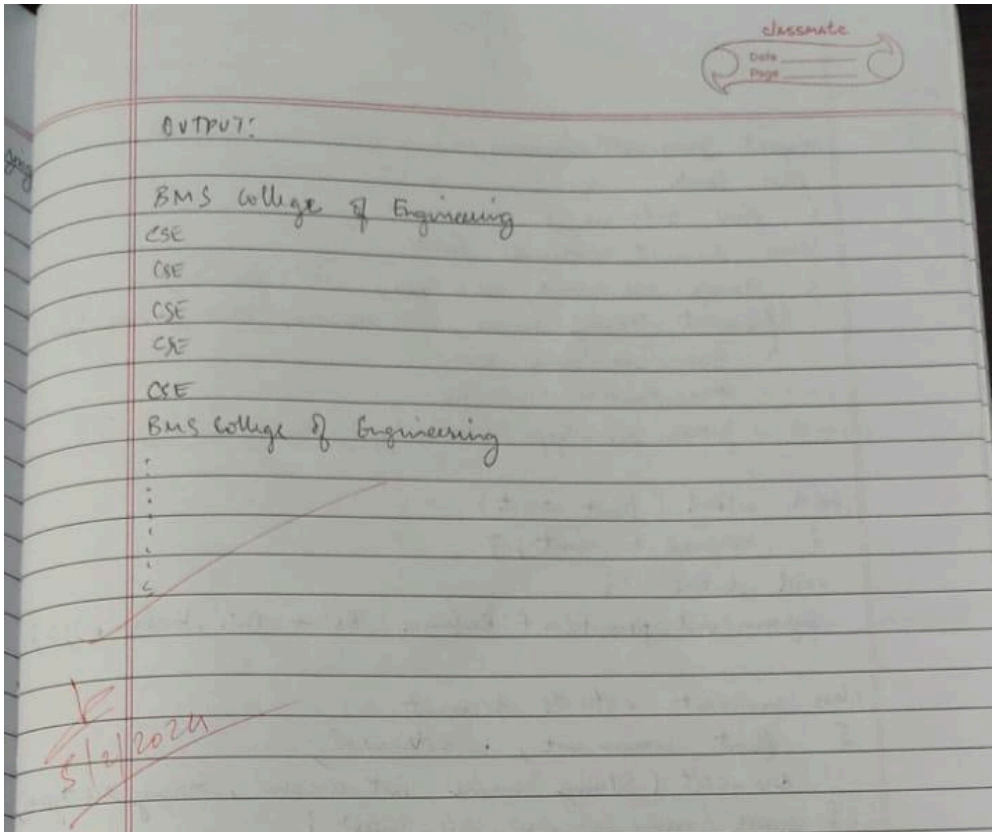
```
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age should be less than Father's age");
        }
        this.sonAge = sonAge;
        System.out.println("Father's Age: " + fatherAge);
        System.out.println("Son's Age: " + sonAge);
    }
}

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter Father's Age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);

            System.out.print("Enter Son's Age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);

        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
```

Q8)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.



8 Write a program which creates 2 threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 2 seconds

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

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

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



CODE :

```
public class ThreadExample {
    public static void main(String[] args) {
        Thread bmsThread = new Thread(() -> {
            while (true) {
                System.out.println("BMS College of Engineering");
                try {
                    Thread.sleep(10000); // 10 seconds
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });

        Thread cseThread = new Thread(() -> {
            while (true) {
                System.out.println("CSE");
                try {
```

```
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });
    bmsThread.start();
    cseThread.start();
  }
}
```

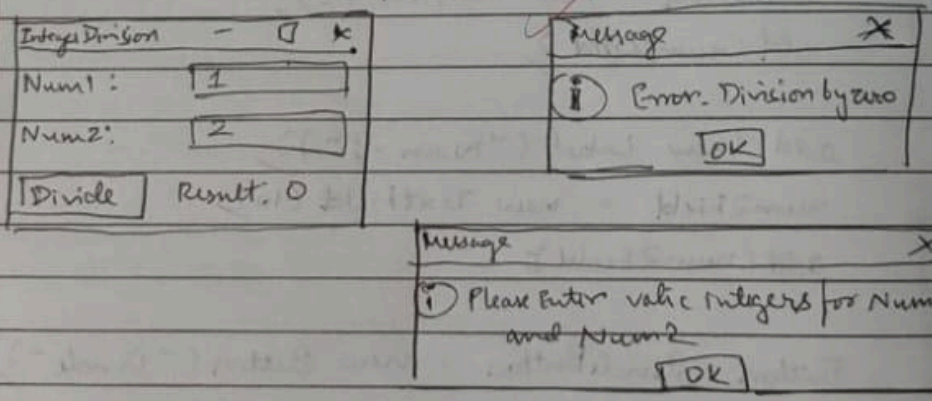
Q9)Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Q9 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.

```
import javax.java.applet.Applet
import java.awt.*;
import java.awt.event.*;

public class IntegerDivisionApplet extends Applet implements
ActionListener {
    private TextField num1Field, num2Field, resultField;
    public void init() {
        setLayout(new GridLayout(4,2));

        add(new Label("Num 1:"));
        num1Field = new TextField(10);
        add(num1Field);

        add(new Label("Num 2:"));
        num2Field = new TextField(10);
        add(num2Field);

        Button divideButton = new Button("Divide");
        divideButton.addActionListener(this);
        add(divideButton)

        add(new Label("Result:"));
        resultField = new TextField(10);
```

```
resultField.setEditable(false);
add(resultField);
}

public void actionPerformed(ActionEvent e){
  try {
int num1 = Integer.parseInt(num1Field.getText());
int num2 = Integer.parseInt(num2Field.getText());

  if(num2 == 0){
    throw new ArithmeticException("Division by zero!");
  }

resultField.setText(String.valueOf(num1/num2));
}catch(NumberFormatException | ArithmeticException ex){
  showStatus("Error: " + ex.getMessage());
}
}
}
```

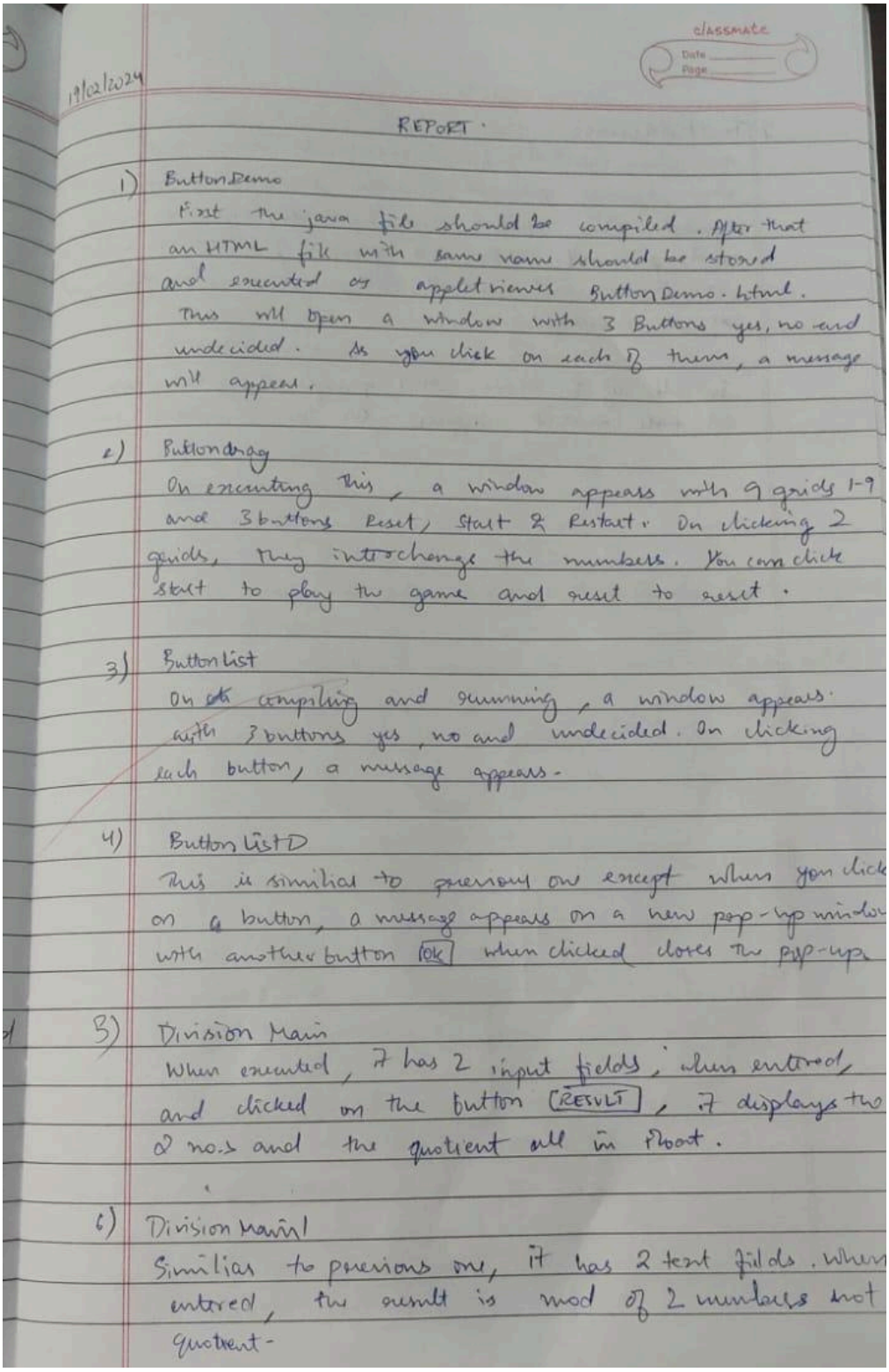
OUTPUT:



CODE :

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

public class IntegerDivisionApplet extends Applet implements ActionListener {
    private TextField num1Field, num2Field, resultField;

    public void init() {
        setLayout(new GridLayout(4, 2));

        add(new Label("Num1:"));
        num1Field = new TextField(10);
        add(num1Field);

        add(new Label("Num2:"));
        num2Field = new TextField(10);
        add(num2Field);

        Button divideButton = new Button("Divide");
        divideButton.addActionListener(this);
        add(divideButton);

        add(new Label("Result:"));
        resultField = new TextField(10);
        resultField.setEditable(false);
        add(resultField);
    }

    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());

            if (num2 == 0) {
                throw new ArithmeticException("Division by zero!");
            }

            resultField.setText(String.valueOf(num1 / num2));
        } catch (NumberFormatException | ArithmeticException ex) {
            showStatus("Error: " + ex.getMessage());
        }
    }}
```

REPORT ON AWT PROGRAMS :



19/02/2024

REPORT.

1) ButtonDemo

First the java file should be compiled. After that an HTML file with same name should be stored and executed as appletviewer ButtonDemo.html. This will open a window with 3 Buttons yes, no and undecided. As you click on each of them, a message will appear.

2) Buttondrag

On executing this, a window appears with 9 grids 1-9 and 3 buttons Reset, Start & Restart. On clicking 2 grids, they interchange the numbers. You can click start to play the game and reset to reset.

3) ButtonList

On ~~ok~~ compiling and running, a window appears with 3 buttons yes, no and undecided. On clicking each button, a message appears.

4) ButtonListD

This is similar to previous one except when you click on a button, a message appears on a new pop-up window with another button [Ok] when clicked closes the pop-up.

5) Division Main

When executed, it has 2 input fields, when entered and clicked on the button [RESULT], it displays the 2 nos and the quotient all in Root.

6) Division Main1

Similar to previous one, it has 2 text fields. When entered, the result is mod of 2 numbers not quotient.

1. TextFieldDemo

This when executed, has 2 text fields Name and Password. When typing password, it hides it and displays * symbol. Once clicked enter, it displays the name and password. When a group of letters is selected in Name and entered click, it is also displayed as a message.

In all of the above AWT programs, one can set title and it appears as the window name.