Krishna Rugved Durbha(1BM22CS131)

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.

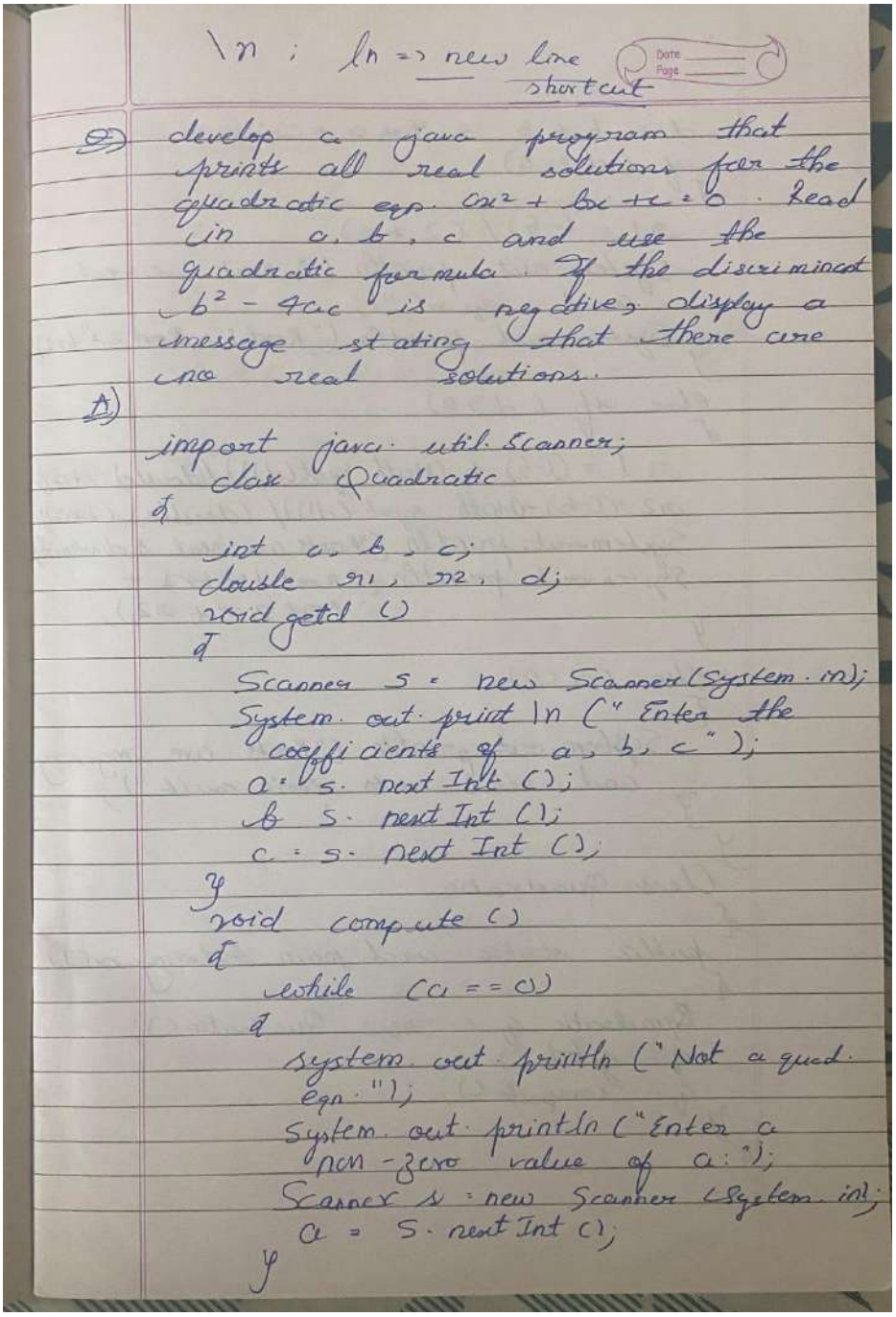
```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{public static void main(String XX[])
 {
    int a,b,c;
    System.out.println("enter the values of a,b,c respectively\n");
    Scanner s1= new Scanner(System.in);
    a = s1.nextInt();
    b = s1.nextInt();
    c = s1.nextInt();
    double d= b*b - 4*a*c ;
    System.out.println("a = " + a +" b = " + b +" c = " + c);
    if(a==0) {System.out.println("not a quadratic equation");}
    else if( d>0)
    {
      System.out.println("the equation has two real and different solutions");
      double r1=(-b + Math.sqrt(d))/(2*a);
      double r2=(-b - Math.sqrt(d))/(2*a);
      System.out.println("r1 = " + r1);
      System.out.println("r2 = " + r2);
    }

   else if(d==0)
    {
      System.out.println("the equation has real and equal solutions");
      double r1= -b/(2*a);
      double r2= -b/(2*a);
      System.out.println("r1 = " + r1);
      System.out.println("r2 = " + r2);
    }

   else if(d<0)
    {
```

```
    System.out.println("the equation has unreal solutions");
   }
 }

}
```



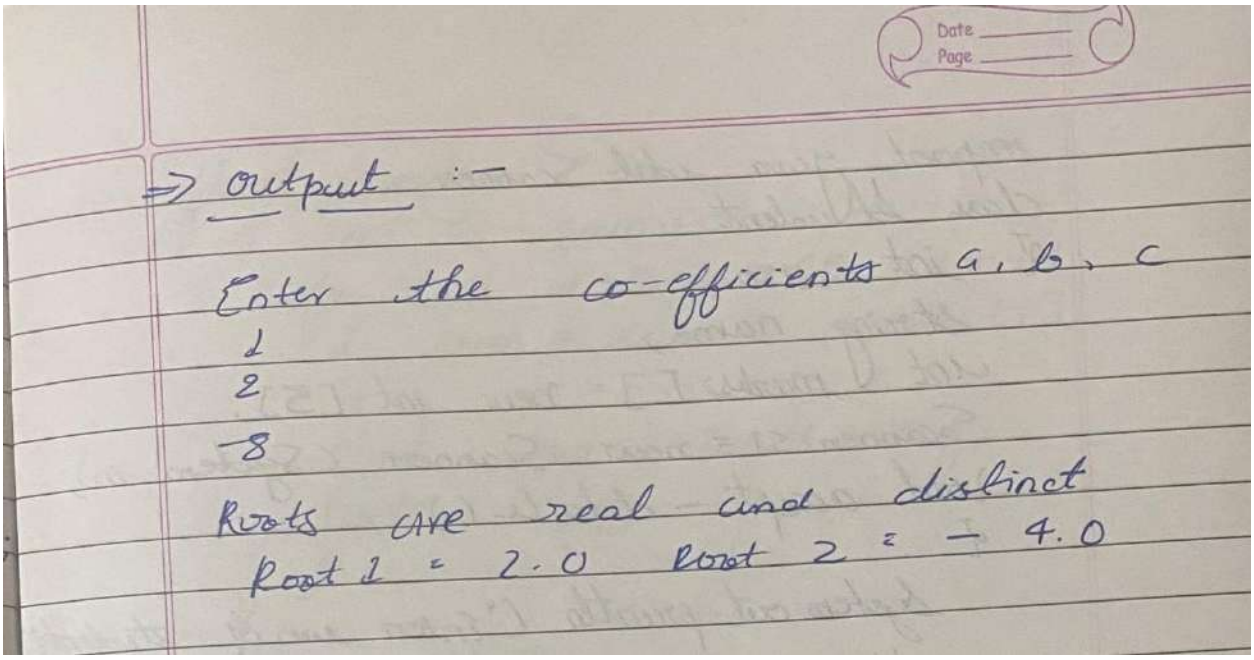
\n ; ln => new line
shortcut

Q) develop a java program that prints all real solutions for the quadratic eqn. $ax^2 + bx + c = 0$. 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.

A)

```
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, c");
     a = s.nextInt ();
     b = s.nextInt ();
     c = s.nextInt ();
   }
   void compute ()
   {
     while (a == 0)
     {
       System.out.println ("Not a quad.
        eqn.");
       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 (" Root 1 = Root 2 =" + 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 & distinct");
    System.out.println (" Root 1 =" + r1 +
                        " Root 2 =" + r2);
}
Else (d < 0)
{
    System.out.println (" Roots are imaginary
      and real roots don't exist ");
}
}
Class Quadratic
{
public static void main (String xx[])
{
    Quadratic q = new Quadratic ();
    q.getd ();
    q.compute ();
}
}
```



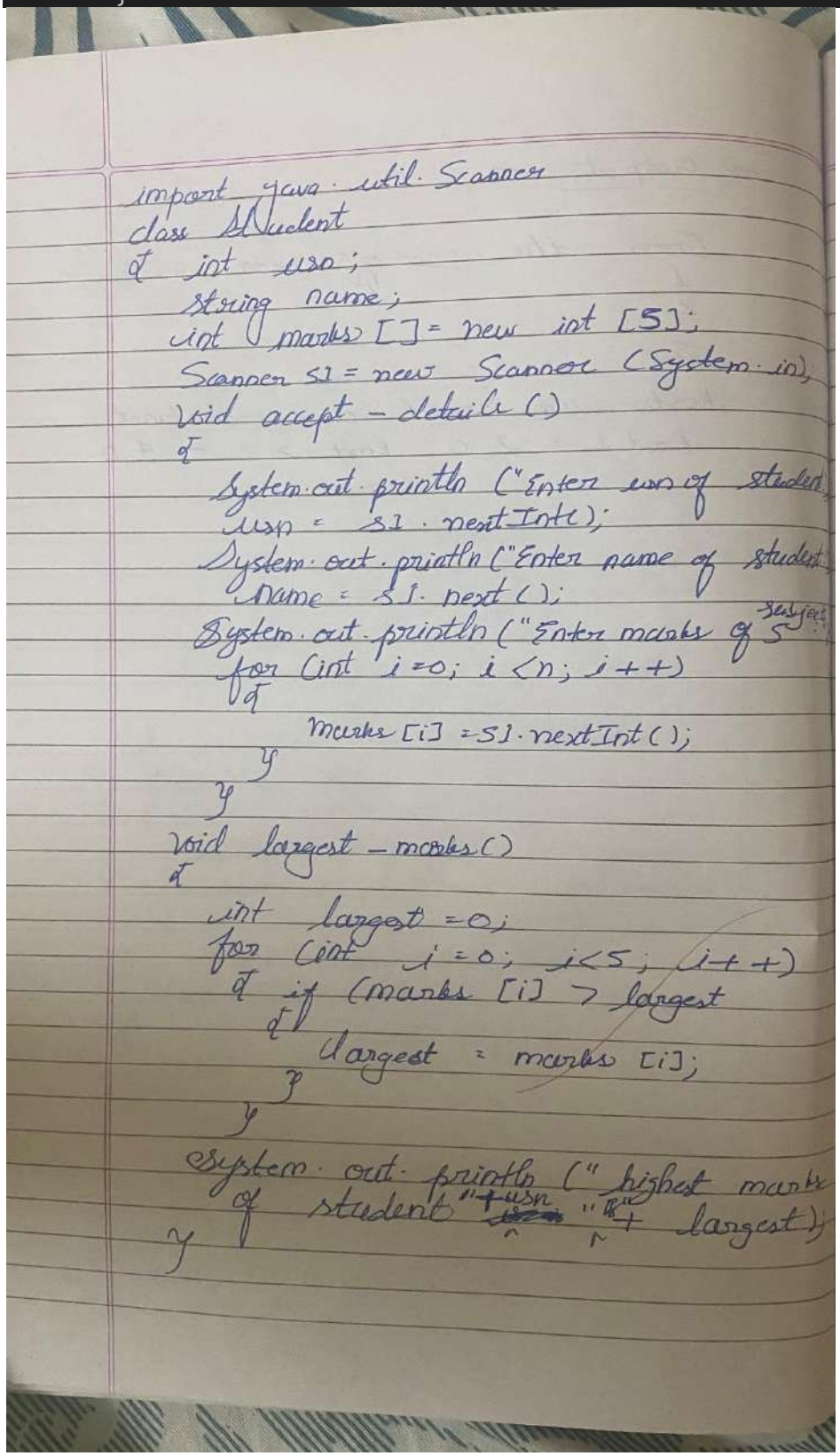
Q2. Develop a Java program to create a class Student with members usn, name, and array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class student{
    String USN , name;
    Scanner S1= new Scanner(System.in);
    int size = S1.nextInt();
    float credits[] =  new float[size];
    float marks[] =  new float[size];
    void accept(){
        USN= S1.next();
        name= S1.next();
        System.out.println("Marks in the following subjects are 1.Maths 2.physics
3.C progm 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
            marks[i]= S1.nextInt();
        }
```

```
        System.out.println("respective credits of subjects are 1.Maths 2.physics
3.C progm 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
        credits[i]= S1.nextInt();
        }
    }
    void display(){
        System.out.println("USN: "+USN+" name: "+name);
        System.out.println("Marks and credits in the following subjects are
1.Maths 2.physics 3.C progm 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" .marks = "+marks[i]+" credits="+credits[i]+"
");
        }
    }
    int gpa(int i){
            if(marks[i]>=90) return 10;
            else if(marks[i]>=80 && marks[i]<=89) return 9;
            else if(marks[i]>=70 && marks[i]<=79) return 8;
            else if(marks[i]>=60 && marks[i]<=69) return 7;
            else if(marks[i]>=50 && marks[i]<=59) return 6;
            else if(marks[i]>=40 && marks[i]<=49) return 5;
            else if(marks[i]>=80 && marks[i]<=89) return 4;
            else return 0;
            }

    float sgpa(){
        float SGPA , sum=0;
        for(int i=0;i<size;i++){
            sum=sum+gpa(i)*credits[i];
        }
        //total credits=20
        SGPA = sum/20;
        return SGPA;
    }
}
    class call{
        public static void main (String[] args) {
            student S1 = new student();
            S1.accept();
            S1.display();
            float Ans;
            Ans = S1.sgpa();
            System.out.println("SGPA of the student is : "+Ans);
```

```
        }
    }
```



```
import java.util.Scanner
class Student
{ int usn;
  String name;
  int marks[] = new int[5];
  Scanner s1 = new Scanner(System.in);
  void accept_details()
  {
    System.out.println("Enter usn of student");
    usn = s1.nextInt();
    System.out.println("Enter name of student");
    name = s1.next();
    System.out.println("Enter marks of 5 subjects");
    for(int i=0; i<n; i++)
    {
        marks[i] = s1.nextInt();
    }
  }
  void largest_marks()
  {
    int largest = 0;
    for(int i=0; i<5; i++)
    { if(marks[i] > largest
      {
        largest = marks[i];
      }
    }
    System.out.println("highest marks of student "+usn+" "+" "+largest);
  }
```

```
float average ()
{ int sum = 0;
  for (int i = 0 ; i < 5 ; i++)
  { sum = sum + marks [i];
  }
  float avg = sum / 0.5f;
  return avg;
  }
}
Class Stu2emo
{
  public static void main (String args[])
{
  System.out.println (" Enter no. of students ");
  Scanner S2 = new Scanner (System.in);
  int n = S2.nextInt ();
  Student S [] = new Student [n],
  for (int i = 0; i < n ; i++)
  {
      S[i] = new Student ();
      S[i].accept_details ();
      S[i].largest_marks ();
  }
  int top = 0;
  float maxavg = 0;
  for (int j = 0; j < n; j++)
  {
    if (S[j].average () > maxavg)
    {
      maxavg = S[j].average ();
      top = s[j] - usn;
    }
  }
  System.out.println (" Topper is "+top);
}}
```

```
import java.util.Scanner
class Student
{
    int usn;
    String name;
    int[] marks = new int[8];
    int[] credit = new int[8];
    int[] credit = new int[8];
    Scanner si = new Scanner(System.in);
    void accept_details()
    {
        System.out.println("Enter student usn:");
        usn = si.nextInt();
        System.out.println("Enter student name:");
        name = si.next();
        System.out.println("Enter marks
        in order of credits:");
        for(int i=0; i<8; i++)
        {
            marks[i] = si.nextInt();
        }
        System.out.println("Enter order
        of credits:");
        for(int i=0; i<8; i++)
        {
            credit[i] = si.nextInt();
        }
    }
}
```

```
void Calculate()
{ for (int i=0; i<8; i++)
  {
    if (marks[i] >= 90)
    {
      credit_points[i] = 10 * credit[i];
    }
    else if (marks[i] >= 80)
    { credit_points[i] = 9 * credit[i];
    }
    else if (marks[i] >= 70)
    { credit_points[i] = 8 * credit[i];
    }
    else if (marks[i] >= 60)
    { credit_points[i] = 7 * credit[i];
    }
    else if (marks[i] >= 50)
    { credit_points[i] = 6 * credit[i];
    }
    else if (marks[i] >= 40)
    { credit_points[i] = 5 * credit[i];
    }
  }
  int sum = 0;
  int count = 0;
  double SGPA;
  for (int j=0; j<8; j++)
  {
    sum = sum + credit_point
    count = count + credit[j]; [cj];
  }
```

```
SGPA = sum / count ;
System.out.println("SGPA is"
                    +SGPA);
    }
}
class Main
{
    public static void main (String
                              args[])
{ System.out.println("Enter no.
        of students:");
    Scanner S = new Scanner(System.
                              in);
    int n = S.nextInt();
    Student s[] = new Student[n];
    for(int i=0; i<n; i++)
    {
        st[i] = new Student();
        st[i].accept_details();
        st[i].calculate();
    }
}
```

9/1/24

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.

```
import java.util.Scanner;
class Book{
    String name, Author;
    int price,num_pages;
    Book(){};
    Book( String name,String Author,int price, int num_pages){
        this.name=name;
        this.Author=Author;
        this.price=price;
        this.num_pages=num_pages;
    }
    void set(){
        System.out.println("enter details of book :");
        Scanner S1= new Scanner(System.in);
        name = S1.next();
        Author = S1.next();
        price = S1.nextInt();
        num_pages= S1.nextInt();
    }
    void get(){
        System.out.println("name: "+name+" Author: "+Author+" price: "+price+"
num_pages: "+num_pages);
    }
    public String toString(){
        return("name: "+name+" Author: "+Author+" price: "+price+" num_pages:
"+num_pages);
    }
}
class bookdemo{
     public static void main (String[] args) {
        int n;
        Scanner S= new Scanner(System.in);
        System.out.print("print n: ");
        n=S.nextInt();
```

```
        Book B[] = new Book[n];
        B[0] = new Book("cant_hurt_me","David_googins",700,360);
        B[0].get();
        for(int i=1;i<n;i++){
        B[i]= new Book();
        B[i].set();    }
    System.out.println(B[1].toString());
    for(int i=2;i<n;i++){
        B[i].get();
        }    }}
```

```
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.
    System.out.println (" Enter book name:"); in);
    this.name = scanner.nextLine();
    System.out.println ("Enter author name:");
    this.author = scanner.nextLine();
    System.out.println ("Enter price:");
    this.price = scanner.nextDouble ();
    System.out.println ("Enter no. 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 ("No. of pages:" + Num
                                        pages);
}
```

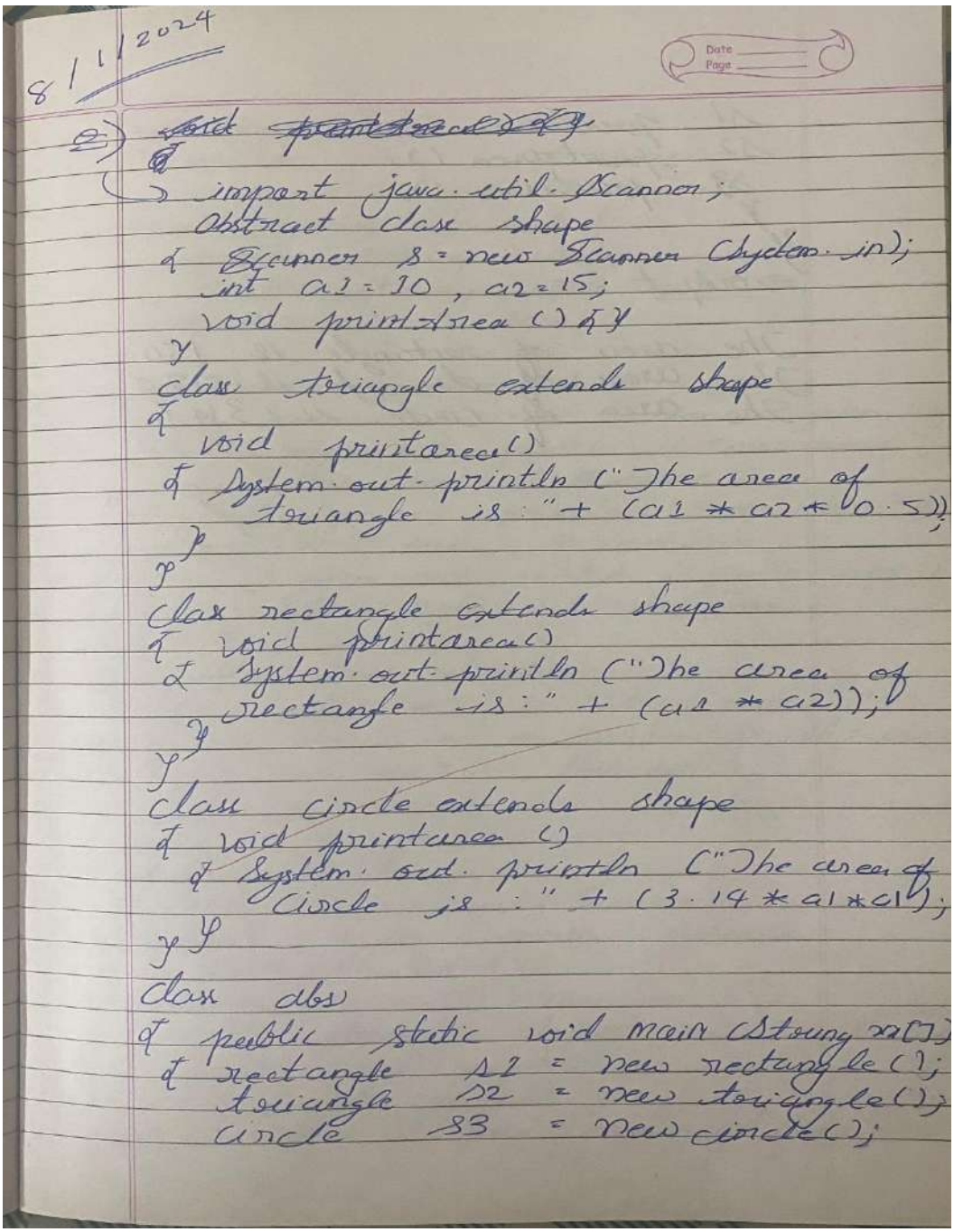
```
public String toString()
{ return "Book Details: \n" + "Name: " +
  name + "\n" + "Author: " + author +
  "\n" + "Price: $" + price + "\n" +
  "Number of Pages: " + numpages; }
}

public class Main {
  public static void main (String[] args)
  { Scanner scanner = new Scanner
                        (System.in);
System.out.println ("Enter the no. of books:");
int n = scanner.nextInt();
Book[] books = new Book[n];
for (int i=0; i<n; i++)
{ System.out.println ("\n Enter details
  for the Book " + (i+1) + ":");
books[i] = new Book ("", "", 0.0, 0);
books[i].setDetails();
}
System.out.println ("\n Details of all
  books:");
for (int i=0; i<n; i++)
{ System.out.println ("\n Book" +
  (i+1) + ":");
  book[i].getDetails();
}
System.out.println ("\n Complete
  details of all books:");
for (int (i)=0; i<n; i++)
{ System.out.println ("\n Book" +
    (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.

```
abstract class shape{
    int a,b;
    abstract void printarea();
}
class rectangle extends shape{
    rectangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of rectangle is : "(a*b));
    }
}
class triangle extends shape{
    triangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of triangle is : "(0.5*a*b));
    }
}
class circle extends shape{
    circle(int x){
        a=x;
    }
    void printarea(){
        System.out.println("area of circle is : "(3.14*a*a));
    }
}
class shapedemo{
    public static void main(String xx[]){
        rectangle r=new rectangle(5,4);
        triangle t=new triangle(5,4);
        circle c=new circle(7);
        r.printarea();
```

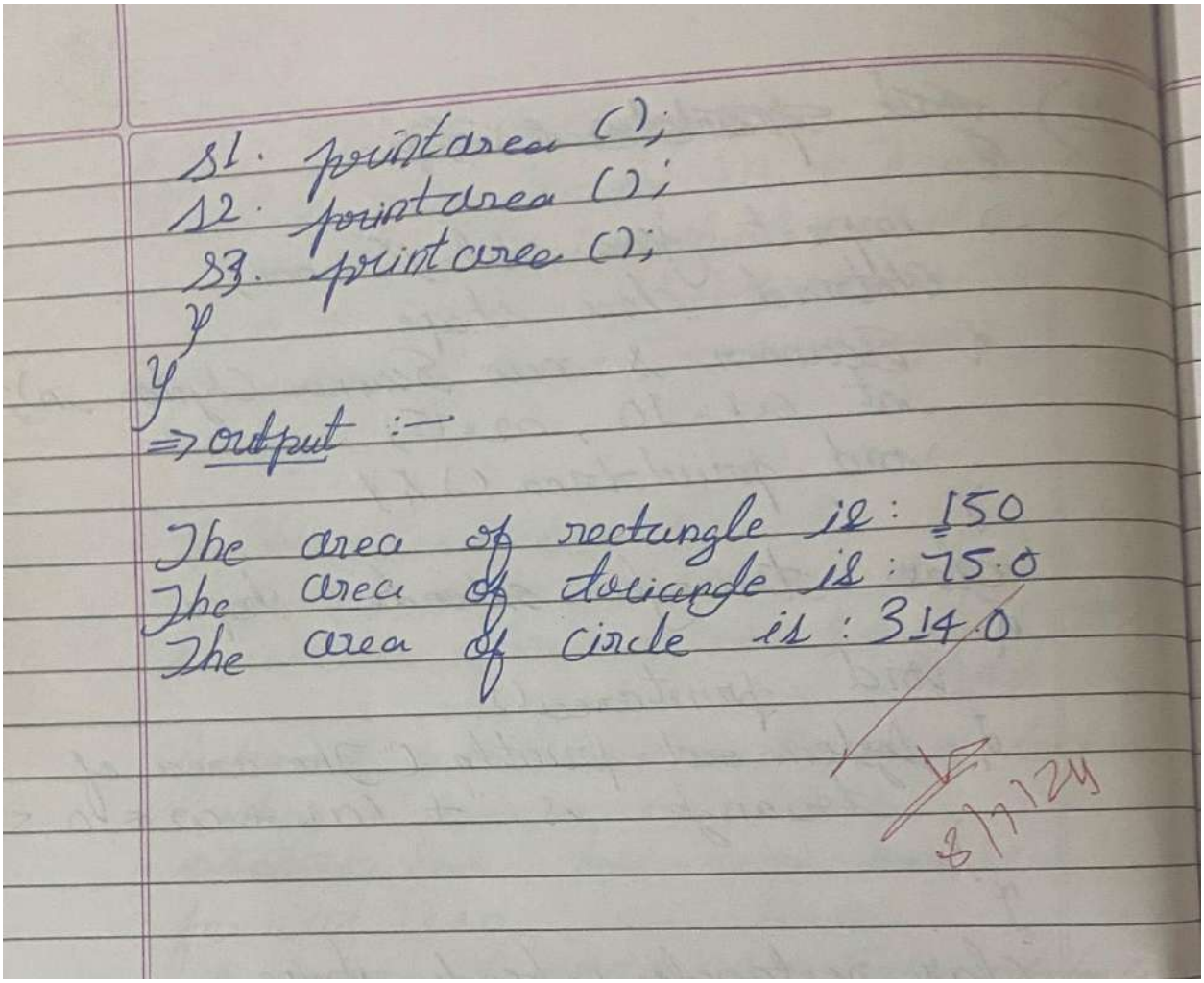
```
        t.printarea();
        c.printarea();
    }
}
```



8/1/2024

2)

```
import java.util.Scanner;
Abstract class shape
{ Scanner s = new Scanner(System.in);
  int a1 = 10, a2 = 15;
  void printArea() { }
}
class triangle extends shape
{
  void printarea()
  { System.out.println("The area of
    triangle is: " + (a1 * a2 * 0.5));
  }
}
class rectangle extends shape
{ void printarea()
  { System.out.println("The area of
    rectangle is: " + (a1 * a2));
  }
}
class circle extends shape
{ void printarea()
  { System.out.println("The area of
    circle is: " + (3.14 * a1 * a1));
  }
}
class abs
{ public static void main(String a[])
  { rectangle s1 = new rectangle();
    triangle s2 = new triangle();
    circle s3 = new circle();
```



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.

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String name, int accNo, String accType, double initialBalance) {
        customerName = name;
        accountNumber = accNo;
        accountType = accType;
        balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful.");
    }

    void displayBalance() {
        System.out.println("Balance: $" + balance);
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    CurAcct(String name, int accNo, String accType, double initialBalance, double
minBal, double charge) {
        super(name, accNo, accType, initialBalance);
        minBalance = minBal;
        serviceCharge = charge;
    }
```

```
    void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }

    void deductServiceCharge() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of $" + serviceCharge + " applied
due to balance below minimum.");
        }
    }
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNo, String accType, double initialBalance, double
interest) {
        super(name, accNo, accType, initialBalance);
        interestRate = interest;
    }

    void calculateInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of $" + interest + " added.");
    }

    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

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

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

        // Creating savings account
        SavAcct savings = new SavAcct("John Doe", 123456, "Savings", 1000, 5); //
5% interest rate

        // Creating current account
        CurAcct current = new CurAcct("Jane Doe", 654321, "Current", 2000, 500,
10); // $500 minimum balance, $10 service charge

        System.out.println("Welcome to our bank!");

        while (true) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4.
Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    System.out.print("Select account (1 for Savings, 2 for
Current): ");
                    int accountChoice = scanner.nextInt();
                    if (accountChoice == 1)
                        savings.deposit(depositAmount);
                    else if (accountChoice == 2)
                        current.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    System.out.print("Select account (1 for Savings, 2 for
Current): ");
                    accountChoice = scanner.nextInt();
                    if (accountChoice == 1)
                        savings.withdraw(withdrawAmount);
                    else if (accountChoice == 2) {
                        current.withdraw(withdrawAmount);
                        current.deductServiceCharge();
                    }
                    break;
                case 3:
```

```
                    System.out.print("Select account (1 for Savings, 2 for
Current): ");
                    accountChoice = scanner.nextInt();
                    if (accountChoice == 1)
                        savings.displayBalance();
                    else if (accountChoice == 2)
                        current.displayBalance();
                    break;
                case 4:
                    System.out.println("Thank you for banking with us!");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```

Q] Display a java program to create a class Bank that maintains 2 kinds of accounts for its customers, Savings and current account. Savings account provides compound interest and withdrawal facilities but no cheque book facility. Current account provides cheque book facility but no interest. Current account holders should also maintain a minimum value / 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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:-

a) accept deposit from customer and update the balance

b) display the balance

c) compute and deposit interest

d) permit withdrawal and update the balance.

```
import java.util.Scanner;
class Account
{ String customerName;
  int accountNumber;
  String accountType;
  double balance;

  Account (String name, int AccNo,
  String accType, double initialBalance)
  { customerName = name;
    accountNumber = accNo;
    accountType = accType;
    balance = initialBalance;
  }
void deposit (double amount)
{ balance += amount;
  System.out.println ("Deposit of
    $" + amount + "successful!");
}
void displayBalance ()
{ System.out.println ("Balance $"
    + amount balance );
}
}
class CurAct extends Account
{ double minBalance;
  double serviceCharge;
  CurAct (String name, int accNo,
  String accType, double initialBalance,
  double minBal, double charge)
{ minBalance = minBal;
  serviceCharge = charge;
}
```

```
void withdraw (double amount)
{ if (balance - amount >= minBalance)
  { balance -= amount;
    System.out.println ("Withdrawal
    of ₹" + amount + "Successful!");
  }
  else
  { System.out.println ("Insufficient
    funds. Withdrawal failed.");
  }
}
void deductServiceCharge()
{ if (balance < minBalance)
  { balance -= serviceCharge;
    System.out.println ("Service Charge of ₹"
    + serviceCharge + "applied due to
    balance below minimum");
  }
}
}

class SavAct extends Account
{ double interestRate;
  SavAct (String name, int accNo,
  String accType, double initialBalance,
  double interest)
  { super (name, accNo, accType,
    initialBalance);
    interestRate = interest;
  }
```

```
void calculateInterest()
{ double interest = balance * interestRate/100;
  balance += interest;
  System.out.println("Interest of $"
    + amount + " added successful!");
}
void withdraw(double Amount)
{ if (balance - amount >= 0)
  { balance -= amount;
    System.out.println("Insufficient
    funds. Withdrawal failed.");
    System.out.println("Withdr-
    awal of $" + amount +
    " Successful!");
  }
  else
  { System.out.println("Insufficient
    funds. Withdrawal failed.");
  }
}

Class Bank
{
public static void main(String[]
  args)
{ Scanner scanner = new Scanner
                    (System.in);
SavAcct Savings = new SavAcct("John Doe",
123456, "Savings", 1000, 5);

CurAcct Current = new CurAcct("Jane Doe",
654321, "Current", 2000, 500, 10);
```

```
System.out.println("welcome to
  our bank");

while (true)
{ System.out.println("\n 1) Deposit\n
    2) Withdraw\n 3) Display Balance
    \n 4) Exit");
  System.out.println("Enter your
    choice :");
  int choice = Scanner.nextInt();
switch (choice)
{ case 1:
    System.out.println("Enter amount
      to deposit :");
    double depositAmount = Scanner.next
    System.out.println("Select      Double()
      Account (1 for Savings,
      2 for current") :");
    int AccountChoice = Scanner.next
                                    Int()
    if (account choice == 1)
        Savings.deposit(depositAmount);
    else if (account choice == 2)
        current.deposit(depositAmount);
    break;

case 2:
    System.out.println("Enter amount
    to withdraw :");
    double withdrawAmount = Scanner.
    System.out.println("      next Double()
      Select Account (1 for savings,
      2 for current):");
```

```
accountchoice = scanner.nextInt();
if (accountChoice == 1)
        savings.withdraw(withdrawAmount);
else if (accountchoice == 2)
    { current.withdraw(withdraw
        Amount);
        current.deductServiceCharge();
    }
break;

Case 3:
        System.out.println("Select Account
        (1 for savings, 2 for current)");
        accountchoice = Scanner.nextInt();
        if (accountchoice == 1)
            savings.displayBalance();
        else if (accountchoice == 2)
            current.displayBalance();
        break;

case 4:
        System.out.println("Thank You!");
        System.exit(0);
default:
        System.out.println("Invalid
                            Choice!");
}
}
}
```

Q6. 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 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.

```
package CIE;
import java.util.Scanner;
public class student
{
    public String USN,name;
    public int semester;
    Scanner S1 = new Scanner(System.in);
    public void set()
    {
        System.out.println("USN of the student is : ");
        USN = S1.next();
        System.out.println("name of the student is : ");
        name = S1.next();
        System.out.println("semester of the student is : ");
        semester = S1.nextInt();
    }
    public void get()
    {
        System.out.println("USN of the student is : "+USN+"name of the student is
: "+name+"semester of the student is : "+semester);
    }
}

package CIE;
import java.util.Scanner;
public class internal extends student
{
    Scanner S1 = new Scanner(System.in);
```

```
    public int internal_marks[]= new int[5];
    public void setcie()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.print("\n"+(i+1)+".");
            internal_marks[i]= S1.nextInt();
        }
    }
    public void getcie()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
        System.out.println((i+1)+"."+internal_marks[i]);
        }
    }
}
package SEE;
import CIE.student;
import java.util.Scanner;
public class external extends CIE.student
{
    Scanner s1 = new Scanner(System.in);
    public int see_marks[]=new int[5];
    public void setsee()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.print("\n"+(i+1)+".");
            see_marks[i]= s1.nextInt();
        }
    }
    public void getsee()
    {
        System.out.println(" 1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.println(i+"."+see_marks[i]);
        }
    }
```

```
}

import java.util.Scanner;
import CIE.student;
import CIE.internal;
import SEE.external;
public class fmarks
{
    public static void main(String XX[] )
    {
        int n;
        System.out.println("enter number of students : ");
        Scanner S1=new Scanner(System.in);
        n=S1.nextInt();
        CIE.student S[]=new student[n];
        CIE.internal I[]= new internal[n];
        SEE.external E[]= new external[n];
        for(int i=0;i<n;i++)
        {
            S[i] = new student();
            I[i]= new internal();
            E[i]=new external();
            System.out.println("Enter details of student : ");
            S[i].set();
            System.out.println("Enter internal marks of student (out of 50) : ");
            I[i].setcie();
            System.out.println("Enter see marks of student (out of 100) : ");
            E[i].setsee();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("details of student are : ");
            S[i].get();
            System.out.println("internal marks of student are : ");
            I[i].getcie();
            System.out.println("see marks of student are : ");
            E[i].getsee();
        }
        //final marks calculation
        for (int i = 0; i < n; i++) {
            System.out.println("Marks of student " + S[i].name + ":");
            for (int j = 0; j < 5; j++) {
                int subjectTotalMarks = I[i].internal_marks[j] +
((E[i].see_marks[j])/2);
```

```
                System.out.println("Subject " + (j + 1) + " marks: " +
subjectTotalMarks);
            }
            System.out.println();
        }


    }
}
```

```
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 m[] = new int[5];
    CIE.student student = new CIE.Student();
    public void accept() {
        student.accept();
        Scanner si = 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 ext. marks:");
    for(int i=0; i<5; i++)
    { System.out.println("Marks of
      sub:"+(i+1)+"="+x[i]);
    }
}
```

```
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<5; 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 i=0; i<5; i++)
            {
                double calc = (c1[i].m[j]
                    + ((c2[i].x[j])/2));
                System.out.println("Final
                    marks are [" + (i+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=father’s age.

```
class wrongageexception extends Exception{
    wrongageexception(String message){
        super(message);
    }
    public String toString(){
        return "wrong age enetered";
    }
}
class father{
    int age;
    father(int age) throws wrongageexception{
        if(age<0){
            throw new wrongageexception("age cannot be negative");
        }
        this.age=age;
    }
}
class son extends father{
    int sonage;
    son(int fatherage, int sonage) throws wrongageexception{
        super(fatherage);
        if(sonage >= fatherage){
            throw new wrongageexception("son age cannot be greater than father
age");
        }
        this.sonage=sonage;
    }
}
class exceptiondemo{
    public static void main(String xx[])
    {
        try{
            father f=new father(40);
            son s=new son(f.age,25);
```

```
            System.out.println("father age :"+f.age);
            System.out.println("son age :"+s.sonage);
        }
        catch(wrongageexception e){
            System.out.println("exception:"+e.toString());
            System.out.println("exception:"+e.getMessage());
        }
    }
}
```

⇒ Exception Handling :-

```
import java.util.Scanner;
class WrongAge exception extends
  Exception { public wrong age
  exception (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 WrongAge Exception
  { super (Fatherage);
  if (SonAge >= fatherAge)
    { throw new WrongAge Exception
      (" 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 ExceptionAtInheritance Demo {
    public static void main(String[] args) {
        Scanner s1 = new Scanner(System.in);
        try {
            System.out.println("Enter father's age:");
            int fatherage = s1.nextInt();
            father f = new father(fatherage);
            System.out.println("Enter son's age:");
            int sonAge = s1.nextInt();
            son s = new son(fatherage, sonage);
        }
        catch (WrongAgeException e)
        { System.out.println("Exception:"
            + e.getMessage());
        }
    }
}
```

- Output :

```
Enter father's age: 40
Enter son's age: 20
father's age: 40
son's age: 20
```

Q8. Write a program to create a two threads one thread displays “BMS college of Engineering” once every ten seconds and another displays “CSE” once every two seconds.

```
class BMSthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("BMS College of engineering");
                Thread.sleep(10000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}
class CSEthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("CSE");
                Thread.sleep(2000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}
class display{
    public static void main(String xx[]){
        thread bms=new thread(new BMSthread);
        thread cse=new thread(new CSEthread);
        bms.start();
        cse.start();
    }
}
```

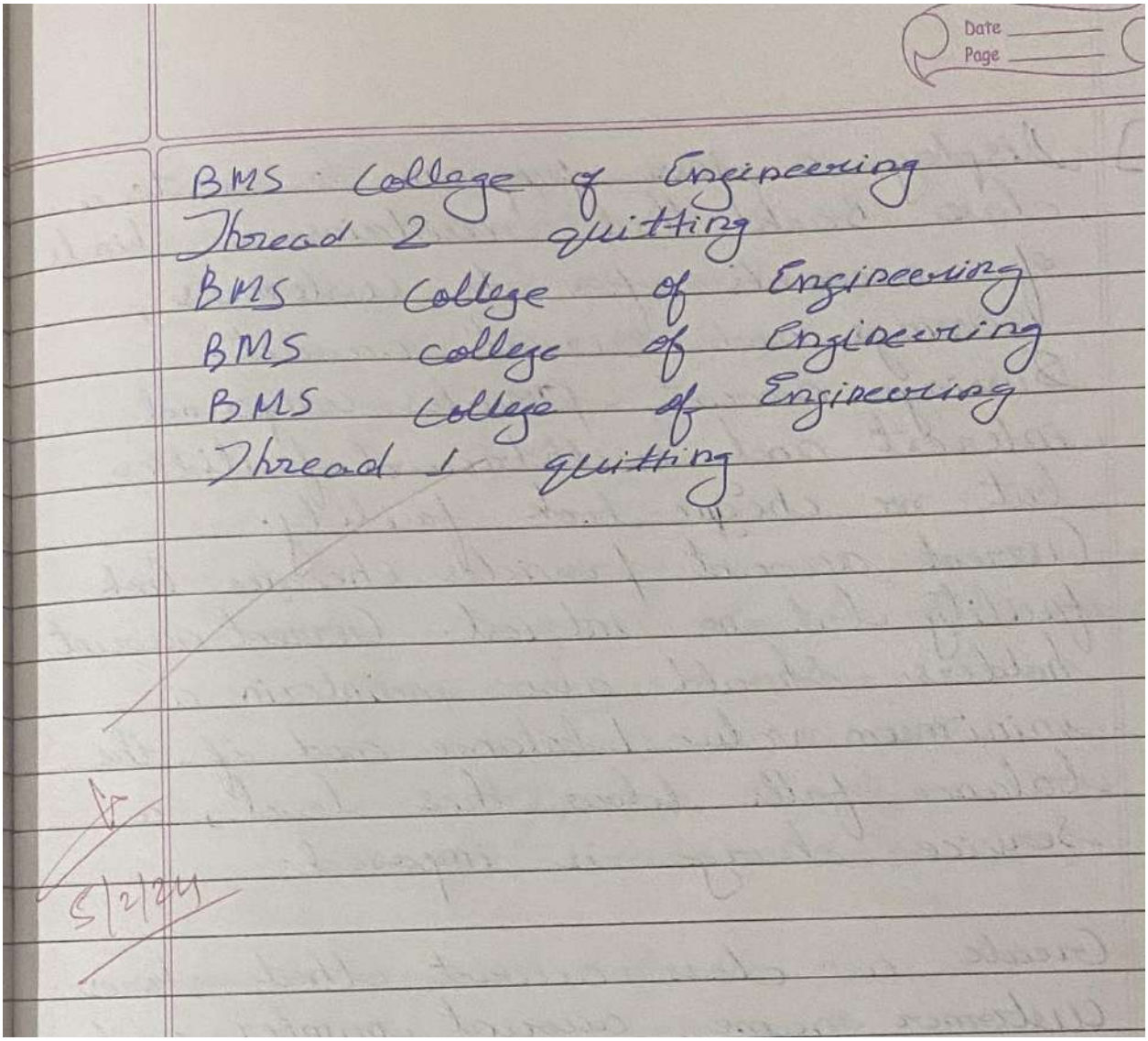
```
class NewThread1 implements Runnable
{ Thread t1;
  NewThread1()
  { t1 = new Thread (this, "Thread1");
    System.out.println ("CT:" + t1);
    t1.start();
  }
public void run()
  { try
    { for (int n = 5; n>0; n--)
      { System.out.println("BMS College of
          Engineering");
        Thread.sleep (1000);
      }
    }
    catch (InterruptedException ie)
    { System.out.println("Thread1
        interrupted");
    }
    System.out.println("Thread1
        quitting");
  }
}
class NewThread2 implements Runnable
{ Thread t2;
  NewThread2()
  { t2 = new Thread (this, "Thread2");
    System.out.println ("CT:" + t2);
    t2.start();
  }
```

```
public void run()
{ try
  { for (int n=5; n>0; n--)
    {
      System.out.println ("CSE");
      Thread.sleep (2000);
    }
  }
  catch (InterruptedException ie)
  { System.out.println ("CSE"
        ("Thread 2 is interrupted");
    }
  System.out.println ("Thread 2 is
    quitting");
  }
}
class MainThread
{ public static void main (String
  { new NewThread1();
    new NewThread2();
  }
}
```

String $$[]

⇒ output :—

CT: Thread [#29, Thread 1, 5, main]
BMS college of Engineering
CT: Thread [#30, Thread 2, 5, main]
CSE
CSE
CSE
CSE
CSE



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 java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class ExceptionDemo extends JFrame implements ActionListener {
 private JTextField t1, t2, t3;
 private JLabel l1, l2;
 private JButton b1, b2;
 public ExceptionDemo() {
 setLayout(new FlowLayout());
 l1 = new JLabel("Num1 :");
 add(l1);
 t1 = new JTextField(5);
 add(t1);
 l2 = new JLabel("Num2 :");
 add(l2);
 t2 = new JTextField(5);
 add(t2);
 t3 = new JTextField(5);
 t3.setEditable(false);
 add(t3);
 b1 = new JButton("Divide");
 add(b1);
 b1.addActionListener(this);
 b2 = new JButton("Clear");
 add(b2);
 b2.addActionListener(this);
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 setSize(300, 150); // Set an appropriate size
 setVisible(true); }
 public void actionPerformed(ActionEvent act) {
 String str = act.getActionCommand();
 if (str.equals("Divide")) {
 try {
 int num1 = Integer.parseInt(t1.getText());
 int num2 = Integer.parseInt(t2.getText());
```

```
 int num3 = num1 / num2;
 t3.setText("" + num3);
 } catch (ArithmeticException e) {
 JOptionPane.showMessageDialog(this, "ArithmeticException: Cannot divide by
zero!");
 } catch (NumberFormatException e) {
 JOptionPane.showMessageDialog(this, "NumberFormatException: Please enter valid
integers for Num1 and Num2." }
 } else {
 t1.setText("");
 t2.setText("");
 t3.setText(""); } }
 public static void main(String[] args) {
 SwingUtilities.invokeLater(new Runnable() {
 public void run() {
 new ExceptionDemo();
 } });}}
```

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class ExceptionDemo
extends JFrame implements ActionListener {
    private JTextField t1, t2, t3;
    private JLabel l1, l2;
    private JButton b1, b2;
    public ExceptionDemo()
    {   setLayout(new FlowLayout());
        l1 = new JLabel("Num 1: ");
        add(l1);
        t1 = new JTextField(5);
        add(t1);
        l2 = new JLabel("Num 2: ");
        add(l2);
        t2 = new JTextField(5);
        add(t2);
        t3 = new JTextField(5);
        t3.setEditable(false);
        add(t3);
        b1 = new JButton("Divide");
        add(b1);
        b1.addActionListener(this);
        b2 = new JButton("Clear");
        add(b2);
        b2.addActionListener(this);

        setDefaultCloseOperation(JFrame.
        EXIT_ON_CLOSE);
        setSize(300, 150);
        setVisible(true); }
```

```
public void actionPerformed (Action
Event act)
{ String str = act.getActionCommand();
  if (str.equals ("Divide"))
  { try {
      int num1 = Integer.parseInt
                 (t1.getText());
      int num2 = Integer.parseInt
                 (t2.getText());
      int num3 = num1 / num2;
      t3.setText ("" + num3);
  }

  catch (ArithmeticException e)
  { JOptionPane.showMessageDialog
    (this, "Arithmetic Exception:
    cannot be divided by zero!");
  }

  catch (NumberFormatException e)
  { JOptionPane.showMessageDialog
    (this, "Number Format Exception:
    Please enter valid integers for
    num1 and num2");
  }

  else { t1.setText ("");
         t2.setText ("");
         t3.setText ("");
  }
```

static

public void main (String [] args)

{ String Utilities ... to be later

(new Runnable ()

{ public void run () {

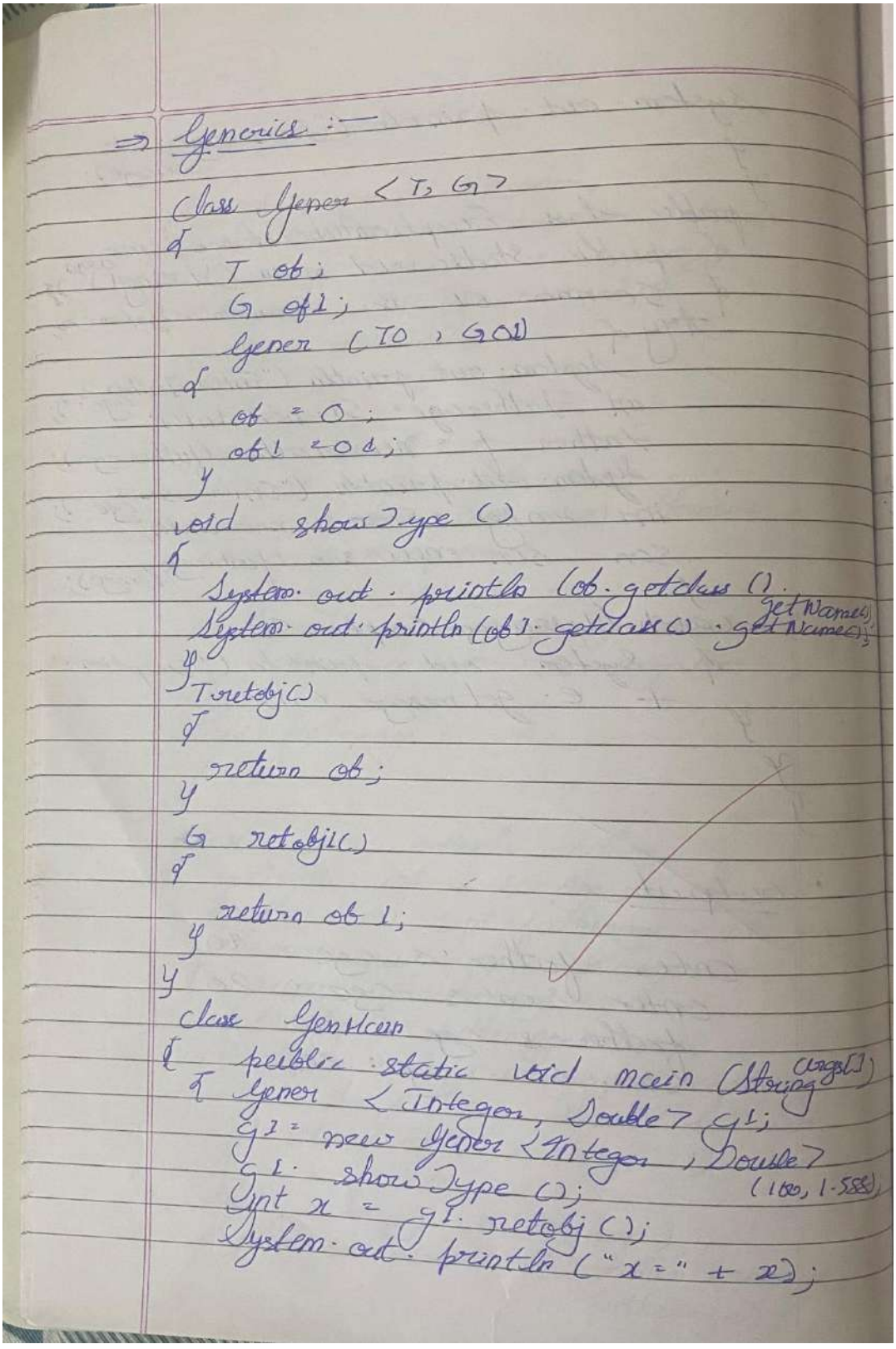
new ExceptionDemo (); }); }

output:

num 1: 12 num 2: 6 2

divide clear

Generics:



⇒ Generics :—

```
Class Gener <T, G>
{
   T ob;
   G ob1;
   Gener (T O, G O1)
   {
     ob = O;
     ob1 = O1;
   }
void showType ()
{
  System.out.println (ob.getclass ().getName());
  System.out.println (ob1.getclass().getName());
}
T retobj()
{
   return ob;
}
G retobj1()
{
   return ob1;
}
}
class GenMain
{ public static void main (String args[])
  { Gener <Integer, Double> g1;
    g1 = new Gener <Integer, Double>
    g1.showType ();                    (100, 1.588)
    int x = g1.retobj ();
    System.out.println ("x =" + x);
```

```
double f = g1.retObj1();
System.out.println("f = " +f);
Gener<Double, Double> g2 = new Gener
    <Double, Double>(253.963, 5963.535);

g2.showType();
double d1 = g2.retObj();
System.out.println(d1 + " " +d1);
Double d2 = g2.retObj1();
System.out.println(d2 + " " +d2);
}
}
```

⇒ output:

```
java.lang.Integer
java.lang.Double
n = 100
f = 1.588
java.lang.Double
java.lang.Double
253.963    253.963
5963.535   5963.535
```