VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

NAVANEETH VN (1BM22CS171)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
October-2022 to Feb-2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Database Management Systems (22CS3PCDBM)" carried out by **NAVANEETH V N(1BM22CS171),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Nandhini Vineeth

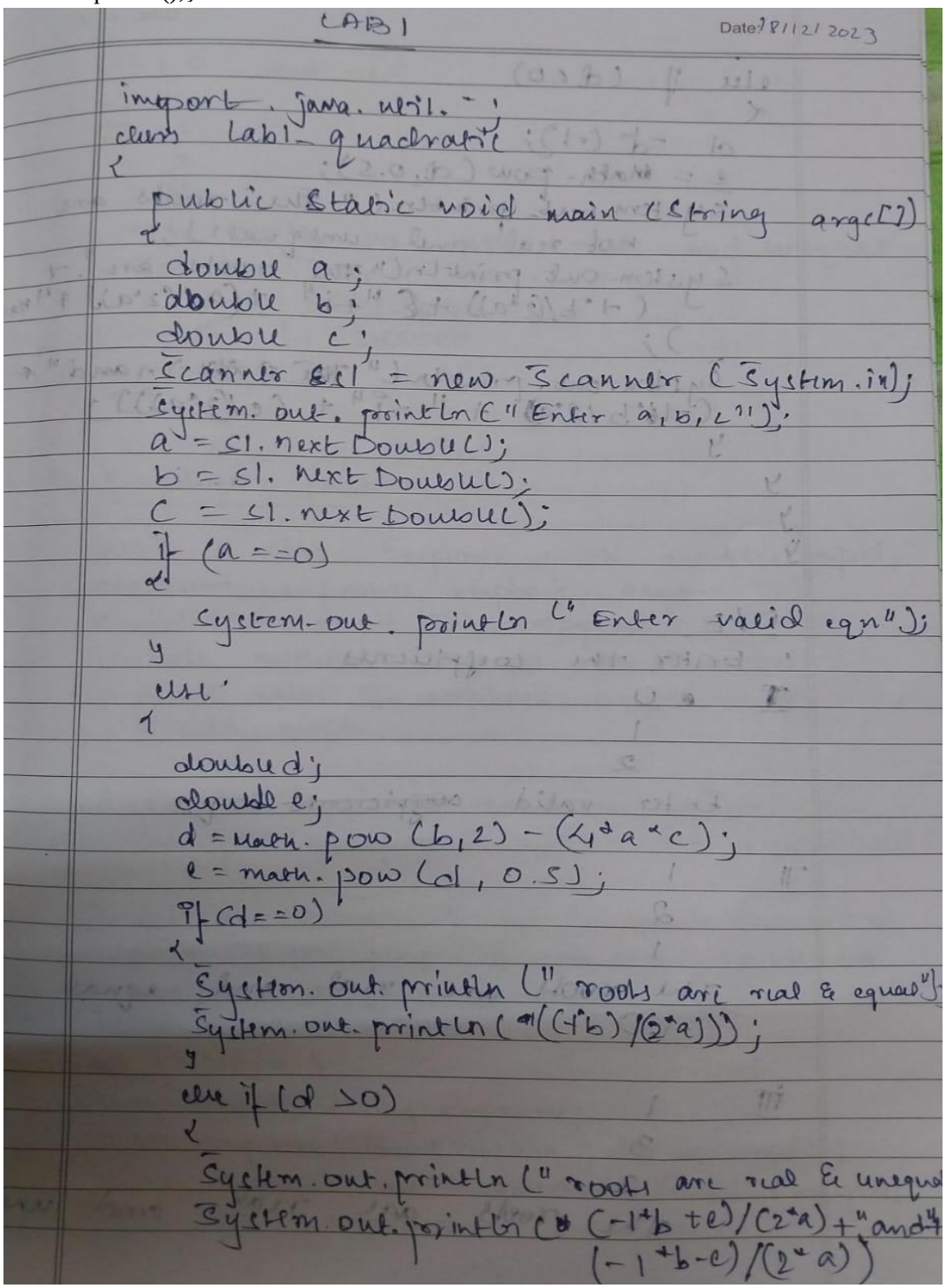
Quadratic Equation

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 discriminate b2-4ac is negative,display a message stating that there are no real solutions.

Code:

```
import java.util.*;
import java.math.*;
class Quadratic
{
	int a,b,c;
	double r1,r2,d;
	void coeff(){
		Scanner s=new Scanner(System.in);
		System.out.println("Enter coeeficients a,b,c");
		a = s.nextInt();
		b = s.nextInt();
		c = s.nextInt();
		d = (b*b) - (4*a*c);
	}
	void evalu(){
		while(a==0){
			System.out.println("Not a QE.");
			System.out.println("Enter non zero coefficient");
			Scanner s = new Scanner(System.in);
			a = s.nextInt();
		}
		if(d==0){
			System.out.println("Roots are real and equal.");
			r1 = (-b)/(2*a);
			System.out.println("Root1=Root2="+r1);
		}
		else if(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);
		}
		else{
			System.out.println("Roots are real and distinct");
			r1 = (-b+(Math.sqrt(d)))/(2*a);
			r2 = (-b-(Math.sqrt(d)))/(2*a);
```

```
                System.out.println("root1= "+r1+"root2= "+r2);
            }
        }
}
class QuadraticEq{
    public static void main(String sx[]){
        Quadratic q = new Quadratic();
        q.coeff();
        q.evalu();}
```



LAB 1

Date: 8/12/2023

```
import java.util.*;
class Lab1_quadratic
{
  public static void main (String args[])
  {
    double a;
    double b;
    double c;
    Scanner sl = new Scanner (System.in);
    System.out.println("Enter a,b,c");
    a = sl.nextDouble();
    b = sl.nextDouble();
    c = sl.nextDouble();
    if (a==0)
    {
      System.out.println("Enter valid eqn");
    }
    else
    {
      double d;
      double e;
      d = Math.pow(b,2) - (4*a*c);
      e = Math.pow(d, 0.5);
      if(d==0)
      {
        System.out.println("roots are real & equal");
        System.out.println("("(-1*b)/(2*a)));
      }
      else if (d>0)
      {
        System.out.println("roots are real & unequa
        System.out.println("("(-1*b+e)/(2*a)+"and"+
                           (-1*b-e)/(2*a))
```

```
else if (d<0)
{
  d = d*(-1);
  e = Math.pow(d,0.5);
  System.out.println("The roots are
      not real and unequal");
  System.out.println("The roots are "+
      (-1*b/(2*a)) +"+"+i" +(e/(2*a)) +"and
  );
  System.out.println("and" +
  (-1*b/2*a) + "-i" + (e/(2*a)));
}
}
}
```

O/P

- Enter the coefficients

I
0
1
2

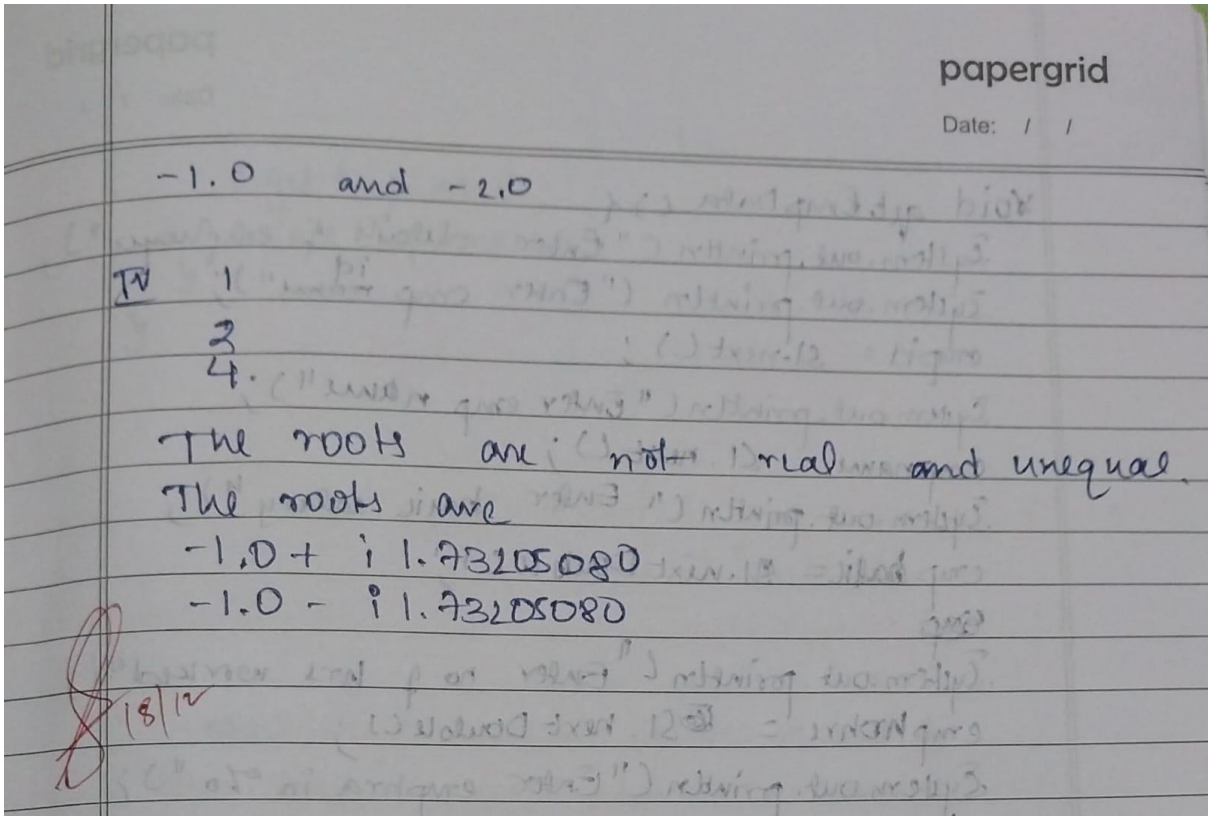
Enter valid ~~coefficients~~ eqn

II
1
2
1

The roots are real and equal
-1.0

III
1
3
2

The roots are real and unea


papergrid
Date: / /
-1.0 and -2.0
IV 1
2
4.
The roots are not real and unequal.
The roots are
-1.0 + i 1.73205080
-1.0 - i 1.73205080
18/12

STUDENT SGPA CALCULATOR

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.

Code:

```
import java.util.Scanner;
class StudentDetails{
	Scanner s1 = new Scanner(System.in);
	int credits[] = new int[5];
	double marks[] = new double[5];
	String name;String usn;double sgpa=0;
	void setDetails(){
		System.out.println("Enter the name of the student:");
		name = s1.next();
		System.out.println("Enter the usn of the student:");
		usn = s1.next();
		System.out.println("Enter the credits and the respective marks chosen by the
		student ");
		for(int i=0;i<5;i++){
			System.out.println("The credits of subject "+(i+1)+": ");
			credits[i] = s1.nextInt();
			System.out.println("The marks obtained for subject "+(i+1)+": ");
			marks[i] = s1.nextDouble();
			System.out.println();
		}
	}
	void displayDetails(){
		System.out.println("Enter the details of the student with name "+name);
		System.out.println("USN: "+usn);
		System.out.println("The credits and markks");
		for(int i = 0;i<5;i++){
			System.out.println("Subject "+(i+1));
			System.out.println("Credits "+credits[i]);
			System.out.println("Marks "+marks[i]);
		}
		this.calcSgpa();
	}
	void calcSgpa(){
		int cg = 0;
		for(int i = 0;i<5;i++){
			if(marks[i]>=90 && marks[i]<100){sgpa = sgpa+(10*credits[i]);}
			if(marks[i]>=80 && marks[i]<90){sgpa = sgpa+(9*credits[i]);}
			if(marks[i]>=70 && marks[i]<80){sgpa = sgpa+(8*credits[i]);}
			if(marks[i]>=60 && marks[i]<70){sgpa = sgpa+(7*credits[i]);}
			if(marks[i]>=50 && marks[i]<60){sgpa = sgpa+(5*credits[i]);}
```

```
                        else{sgpa = sgpa+(3*credits[i]);}
                        cg = cg+credits[i];
                }
                sgpa = (sgpa/(cg*10))*5;
                System.out.println("The SGPA obtained by "+name+" is "+sgpa);
        }
}
class Student{
        public static void main(String args[]){
                StudentDetails a = new StudentDetails();
                a.setDetails();
                a.displayDetails();
        }
}
```

Monday
01/01/2024

LAB 2

Q

=>
```
import java.util.Scanner;
class StudentDetails {
    String usn; Scanner s1 = new Scanner(System.in);
    String name;
    int credits[] = new int[5];
    double marks[] = new double[5];
    double sgpa=0;
    void getDetails(){
        System.out.println("Enter usn of student");
        usn = s1.next();
        System.out.println("Enter name of student");
        name = s1.next();
        System.out.println("Enter Credits and marks of the subjects chosen by student");
        for(int i=0; i<5; i++){
            System.out.println("Credits of subject:"+(i+1));
            credits[i] = s1.nextChar();
            System.out.println();
            System.out.println("marks of subject"+(i+1)+"is");
            marks[i] = s1.nextDouble();
        }
    }
    void displayDetails(){
```

```
System.out.println("The details of
    Student " + name );
System.out.println("USN: " + usn );
for(int i=0; i<5; i++){
    System.out.println("subject " + (i+1));
    System.out.println("credits:" + credits[i]);
    System.out.println("marks:" + marks[i]);
}
this.calcSgpa();
}
void calcSgpa(){
    int cg=0;
    for(int i=0; i<5; i++){
        if(marks[i]>=90 && marks[i]<100)
        { sgpa = sgpa + 10* credits[i];}
        else if(marks[i]>=80 && marks[i]<90)
        { sgpa = sgpa + 9* credits[i]; }
        else if(marks[i]>=70 && marks[i]<80)
        { sgpa = sgpa + 8* credits[i]; }
        else if(marks[i]>=60 && marks[i]<70)
        { sgpa = sgpa + 7* credits[i]; }
        else if(marks[i]>=50 && marks[i]<40)
        { sgpa = sgpa + 6* credits[i];}
        else
        { sgpa = sgpa + 3* credits[i]; }
    cg = cg + credits[i];
    }
    sgpa = (sgpa/(cg*10))*5;
    System.out.println("The SGPA obtained by "
    + name + " is " + sgpa);
}
}
```

```
class Student {
    public static void main (String args[])
        StudentDetails A = new StudentDetails;
        A.setDetails();
        A.DisplayDetails();
}
```

O/P

Enter the usn of student : 171CS
Enter name of student : Navaneeth VN
Enter credits and marks of the subjects chosen by the student
Credits of subject 1 : 4
Mark of the subject 1 : 90
: 4
: 80
: 3
: 94
: 3
: 97
: 1
: 93

The details of student Navaneeth VN
USN : 171CS
Subject 1
Credits 4
Marks 90
Subject 2
Credits 4
Marks 80

Subject 3
credit 3
marks 94
Subject 4
Credits 3
marks 97
Subject 5
credits 1
marks 63
The SGPA obtained by Navaneeth VN is 6.23

Q) BOOK problem

⇒ class Book {

→

Enter usn of student: 1715
Enter name of student: Navaneeth VN
Enter credits & mark of the subject
credits of subject: a
Error
Exception in thread "main" java.util.mismatch Exception

Book Problem

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 andget the details of the objects. Include a toString( ) method that could display thecomplete details of the book. Develop a Java program to create n book objects.

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:\n" +
		"Name: " + name + "\n" +
		"Author: " + author + "\n" +
		"Price: $" + price + "\n" +
		"Number of Pages: " + numPages;
	}
}
public class Books {
	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();
        }
        System.out.println("\nComplete details of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
        }
    }
}
```

Date: / /

- Book problem:

```
import java.util.Scanner;
class Book {
    String name;
    String author;
    double price;
    int num_pages;
    Scanner sl = new Scanner(System.in);
    Book() { }
    Book(String n, String author, double price, int
            num_pages)
    { this.name = n;
      this.author = author;
      this.price = price;
      this.num_pages = pages;
    }
    void setDetails()
    {
    void setName()
    { System.out.println("Enter the Name of the
                book");
      this.name = sl.next();
    }
    void setAuthor()
    { System.out.println("Enter the Name of the
                author");
      this.Author = sl.next()
    }
    void setPrice()
    { System.out.println("Enter price of book");
      this.price = sl.nextDouble();
    }
```

```
void setNumPages(){
System.out.println("Enter no of pages");
this.num_pages = sl.nextLine();
}
void getDetails(){
System
String
void
getName()
{ return name;}
void
String getAuthor(){ return Author;}
double getPrice(){ return price;}
int getPages(){ return num_pages;}

Public String toString()
{
return (getName() + " " + getAuthor() + " " +
getPrice() + " " + getNumPages());
}
}
class main{
public static void main(String args[]){
Book B[] = new Book[2];
Book B1 = new Book("XYZ", "ABC", 1200, 200);
for(int i=0; i<2; i++)
{ B[i]. setName(); B[i] = new Book[];
B[i]. setAuthor();
B[i]. setPrice();
B[i]. setNumPages();
B[i].
System.out.println(B[i]);
}
}
}
```

- Output:

Enter Name of Book:
PQR
Enter Name of Author:
UVW
Enter price of Book:
200.20
Enter no of pages:
100
PQR UVW 200.20 100
Enter name of Book
PITI
Enter name of Author
UXZ
Enter price of Book:
120
Enter no. of pages:
200
PITI UXZ 120 200

- Error

Enter name of Book
PPS
Enter name of Author
XYZ
Enter price of Book
AAC
↳ error

811

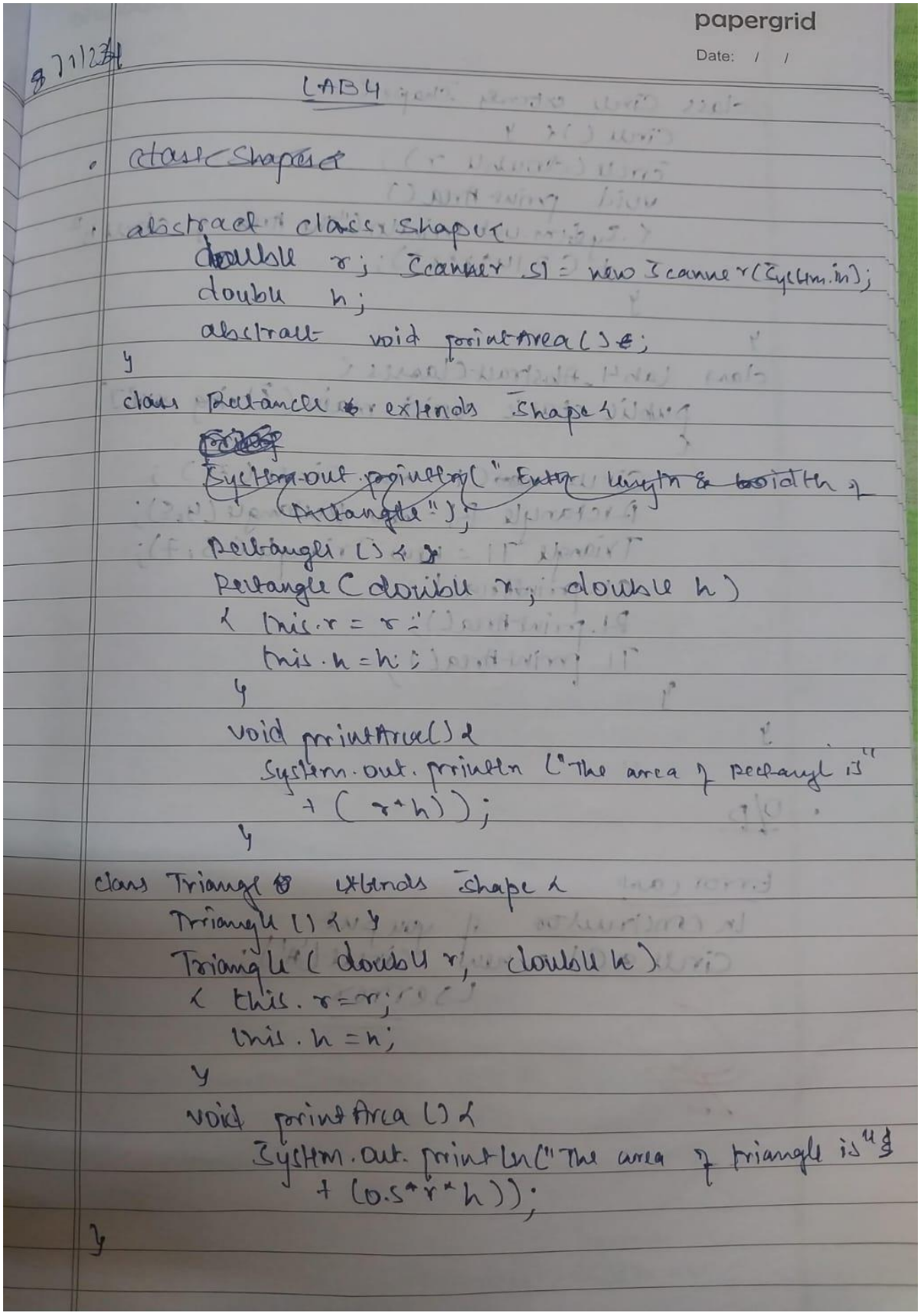
Shape Problem

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.

Code:

```
abstract class Shape {
   int a;
   int b;
   abstract void printArea();
}
class Rectangle extends Shape {
   public Rectangle(int length, int width) {
      super(length, width);
   }
      void printArea() {
      int area = a * b;
      System.out.println("Area of Rectangle: " + area);
   }
}
class Triangle extends Shape {
   public Triangle(int base, int height) {
      super(base, height);
   }
   void printArea() {
      double area = 0.5 * a * b;
      System.out.println("Area of Triangle: " + area);
   }
}
class Circle extends Shape {
   public Circle(int radius) {
      super(radius, radius);
   }
      void printArea() {
      double area = 3.14* a * a;
      System.out.println("Area of Circle: " + area);
   }
}
public class Main {
        public static void main(String[] args) {
                Rectangle rectangle = new Rectangle(5, 10);
                Triangle triangle = new Triangle(4, 6);
                Circle circle = new Circle(7);
                rectangle.printArea();triangle.printArea();circle.printArea();
        }
```

}



LAB 4

```
abstract class Shapes
{
    double r; Scanner sc = new Scanner(System.in);
    double h;
    abstract void printArea();
}
class Rectangle extends Shapes
{
    Rectangle() { }
    Rectangle(double r, double h)
    { this.r = r;
      this.h = h;
    }
    void printArea()
    {
      System.out.println("The area of rectangle is"
        + (r*h));
    }
class Triangle extends Shapes
{
    Triangle() { }
    Triangle(double r, double h)
    { this.r = r;
      this.h = h;
    }
    void printArea()
    {
      System.out.println("The area of triangle is"
        + (0.5*r*h));
}
```

```
class Circle extends Shape {
    Circle() { }
    Circle(double r);
    void printArea()
    { System.out.println("The Area of Circle is "
      + (3.14*r*r));
    }
}
class Lab4_AbstractClasses {
    public static void main(String xx[])
    {
        Circle C1 = new Circle(6.4);
        Rectangle R1 = new Rectangle(4,5);
        Triangle T1 = new Triangle(8,7);
        C1.printArea();
        R1.printArea();
        T1.printArea();
    }
}
```

O/P

Error case

In constructor if you put

Circle C1 = new Circle('A');

↳ error

Bank Problem

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

Code:

```
import java.util.Scanner;
class CanaraBank{
	String customername;
	String accountnumber;
	String type;
	double balance;
	CanaraBank(String customername,String accountnumber,String type){
		this.customername=customername;
		this.accountnumber=accountnumber;
		this.type=type;
	}
	void minimumBalance(){
		if (this.balance<1000){
			System.out.println("You do not fllow rules of minimum balance so 20 rupees
			is deducted");
			balance=balance-20;
		}
	}
}
class SavingsAccount extends CanaraBank{
	double moneyin;
	double interest;
	double depositperiod;
	SavingsAccount(String customername,String accountnumber,String type,double
	moneyin,double interest,doubledepositperiod){
		super(customername,accountnumber,type);
		this.moneyin=moneyin;
		this.interest=interest;
		this.depositperiod=depositperiod;
	}
	void displayBalance(){
```

```
                balance = moneyin*(1+depositperiod*interest);
                System.out.println("The amount in bank after "+ depositperiod + "is "+ balance);
        }
        void withdraw(Double with){
                balance = balance-with;
                this.minimumBalance();
                System.out.println("The amounjt left in bank savings account is "+balance);
        }
        void deposit(){
                System.out.println("ENter the amount you wqant to deposit");
                double deposit; Scanner s1 =new Scanner(System.in);
                deposit=s1.nextDouble(); balance = balance + deposit; this.minimumBalance();
                System.out.println("The amount in bank savings account is "+balance);
        }
}
class CurrentAccount extends CanaraBank{
        double moneyin;
        CurrentAccount(String customername,String accountnumber,String type,double moneyin){
                super(customername,accountnumber,type);
                this.moneyin=moneyin;
                balance=moneyin;
                this.minimumBalance();
        }
        void CheckFacilities(){
                System.out.println("This account has Check facilities");
        }
        void depositThroughCheck(int money){
                balance = balance+money;
                this.minimumBalance();
                System.out.println("The new balance is "+balance);
        }
        void displayBalance(){
                System.out.println("The amount in bank in"+this.type +" account after " + "is "+
                this.balance);
        }
}
class Bank {
        public static void main(String args[]){
                SavingsAccount A = new SavingsAccount("Navaneeth","1000","Savings",900,0.1,1);
                A.displayBalance();
                A.withdraw(20.000);
                A.deposit();
                CurrentAccount B = new CurrentAccount("Monish","1001","Current",1700);
                B.CheckFacilities();
                B.depositThroughCheck(500);
                B.displayBalance();
        }
}
```

Bank problem

```
import java.util.Scanner;

class CanaraBank {
    String customername;
    String accountnumber;
    String type;
    CanaraBank(String Customername, String
                accountnumber, String type){
        this.Customername = customername;
        this.accountnumber = accountnumber;
        this.type = type;
    }
    void minimumBalance(){
        if(this.Balance<1000){
            System.out.println("You dont follow rule
            of minimum balance, so 20 Rs is
            deducted");
            balance = balance -20;
        }
    }
}
class SavingsAccount extends CanaraBank {
    double moneyin; double interest;
    double depositperiod;
    SavingAccount(String customername, String
        accountnumber, String type, double moneyin,
        double interest, double depositperiod){
        super(customername, accountnumber, type);
        this.moneyin = moneyin;
        this.interest = interest;
        this.depositperiod = depositperiod;
    }
```

```
void displayBalance() {
    balance = balance*(1+depositPeriod*interest);
    System.out.println("The amount in the bank after"+
        depositPeriod+" is"+balance);
}
void withdraw(double with){
    balance = balance - with;
    this.minimumBalance();
    System.out.println("The amount left in
        savings bank account is"+balance);
}
void deposit(){
    System.out.println("Enter the amount you want
        to deposit");
    double deposit;
    Scanner s1 = new Scanner(System.in);
    deposit = s1.nextDouble();
    balance = balance + deposit;
    this.minimumBalance();
    System.out.println("New balance is"+balance);
}
}
class CurrentAccount extends CanaraBank{
    double moneyin;
    CurrentAccount(String customername, String accountnumber,
        String type, double moneyin){
        super(customername, accountnumber, type);
        this.moneyin = moneyin;
        balance = moneyin;
        this.minimumBalance();
    }
    void checkFacilities(){
        System.out.println("The account has checkbook
        }
```

```
void depositThroughCheck(int money){
    balance = balance + money;
    this.minimumBalance();
    System.out.println("The new balance is" + bal
}
void displayBalance(){
    System.out.println("The amount in current
        account is" + this.balance);
}
}

class Bank{
    public static void main(String args[]){
        SavingAccount A = new SavingsAccount
            ("Navaneeth MN", "1000", "saving", 900.0);
        A.displayBalance();
        A.withdraw(20.00);
        A.deposit();
        CurrentAccount B = new CurrentAccount
            ("Nimish", "1001", "current", 1700);
        B.checkFacilities();
        B.depositThroughCheck(500);
        B.displayBalance();
    }
}
```

19/2

Output:

```
The amount in bank after 10 is 950.000001
Enter the amount you want to deposit
10
You don't follow minimum balance, so ₹20 deduct
The amount in savings account 940.0000001
The account has cheik facilities.
The new balance is 2200.0000
```

Student Internals And Externals Problem

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.

Code:

```
package CIE;
public class Student{
	public String USN;
	public String Name;
	public int sem;
	public Student(String a,String b,int s){
		USN = a;
		Name = b;
		sem = s;
	}
}

package CIE;
import CIE.Student;
public class Internals extends CIE.Student{
	public int marks[] = new int[5];
	public Internals(String s,String n,int x, int m[]){
		super(s,n,x);
		marks = m;
	}
}

package SEE;
import CIE.Student;
public class External extends CIE.Student{
	public int fy[] = new int[5];
	public External(String s,String n,int sw,int fy[]){
		super(s,n,sw);
		this.fy = fy;
	}
}

import java.util.*;
import java.lang.*;
import CIE.Student;
import CIE.Internals;
import SEE.*;
```

```
public class MainClass{
	public static void main(String args[]){
		int fm = 0;
		Scanner s = new Scanner(System.in);
		System.out.println("Enter the number of students:");
		int n = s.nextInt();
		Internals [] im = new Internals[n];
		External [] em = new External[n];
		Student [] stu = new Student[n];
		for(int i = 0;i<n;i++){
			System.out.println("enter the sdetails of the student "+(i+1));
			System.out.println("ENter the name:");
			String name = s.next();
			System.out.println("Enter usn");
			String usn = s.next();
			System.out.println("Enter the sem");
			int sq =  s.nextInt();
			int imarks[] = new int[5];
			int emarks[] = new int[5];
			System.out.println("ENter the internal markks:");
			for(int j = 0;j<5;j++){
				System.out.println("Enter the internal marks of "+(j+1));
				imarks[j] = s.nextInt();
			}
			for(int j = 0;j<5;j++){
				System.out.print("Enter the external marks of "+(j+1));
				emarks[j] = s.nextInt();
			}
			System.out.println();
			stu[i] = new Student(usn,name,sq);
			im[i] = new Internals(usn,name,sq,imarks);
			em[i] = new External(usn,name,sq,emarks);
		}
		System.out.println("Final marks obtained");
		for(int i = 0;i<n;i++){
			System.out.println("Student"+(i+1)+":");
			System.out.println("Name"+stu[i].Name+"\n"+"USN:"+stu[i].USN+"\n"+"Se
			m"+stu[i].sem);
			for(int j =0;j<5;j++){
				fm = fm +im[i].marks[j]+em[i].fy[j];
				System.out.println("Final marks of course:"+(j+1)+":"+fm);
				fm = 0;
			}
		}
	}
}
```

22/10/2024 LAB 5

papergrid
Date: / /

- Student.java:

```
package CIE;
public class Student
    public String USN, Name;
    public int sem;
    public Student( String USN, String Name,
                    int sem){
        this.USN = USN;
        this.Name = Name;
        this.sem = sem;
    }
}
```

- Internal.java:

```
package CIE;
public class Internals extends Student {
    public int marks[] = new int[5];
    public Internals( String USN, String Name, int sem,
                      int [] m)
    {
        super( USN, Name, sem); this.marks = m;
    }
}
```

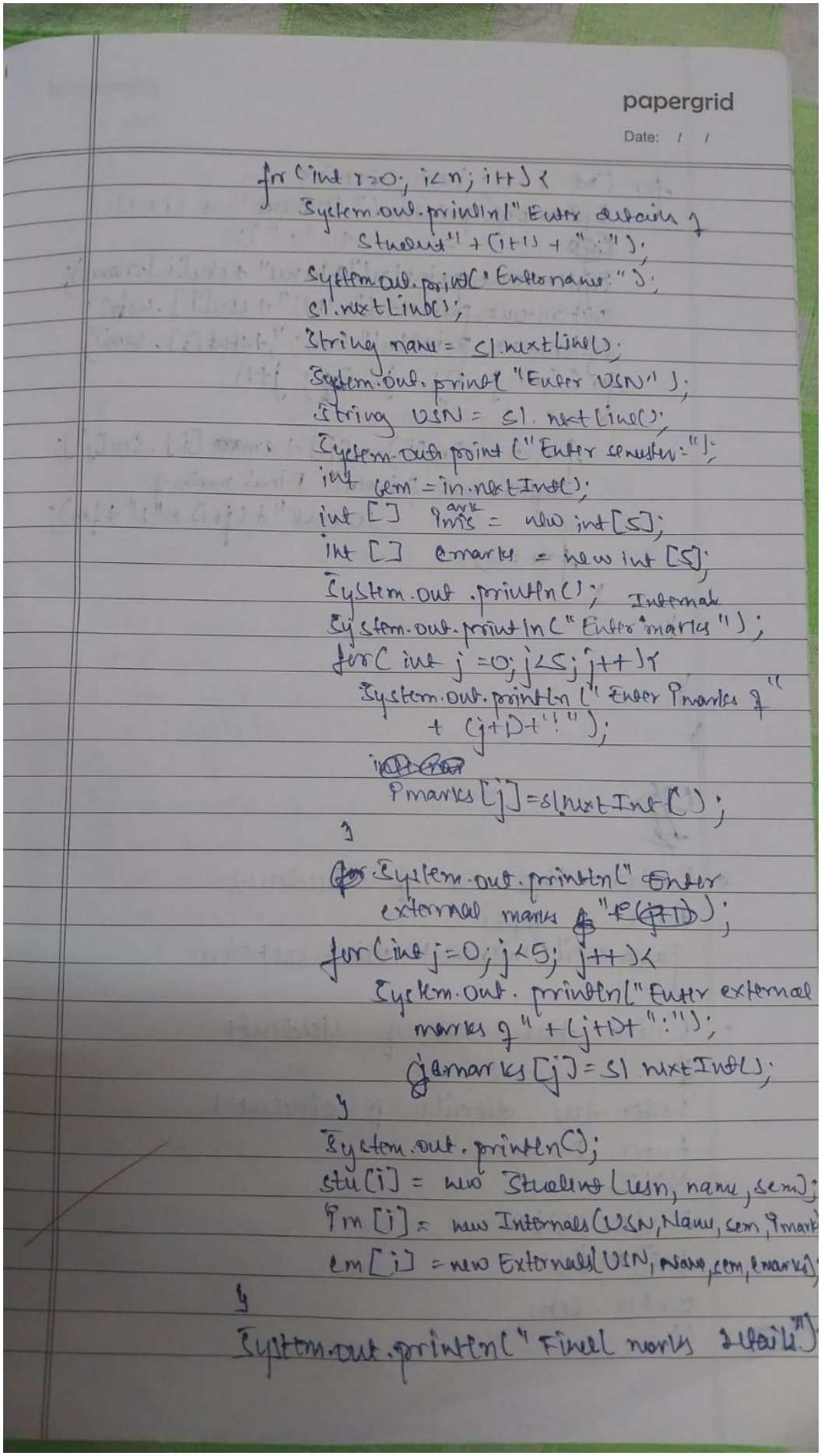
External.java

```
package SEE;
import CIE.Student;                    cil.
public class External extends Student {
    public int [] = new int[5];
                    fy
    public External(
        String USN, String Name, int Sem,
        int[] fy)
    {
        super(usn, Name, sem);
        this.fy = fy;
    }
}
```

mainClass.java:

```
import java.util.*;
import CIE.Student;
import CIE.Externals;
import SEE.*;
public class mainClass {
    public static void main (String args[]) {
        int fm = 0;
        Scanner s1 = new Scanner (System.in);
        System.out.println ("Enter the no
                    of students:");
        int n = s1.nextInt();
        Internal [] im = new Internals[n];
        External[] em = new External[n];
        Student [] stu = new Student[n];
```

980



```
for(int i=0; i<n; i++){
    System.out.println("Enter details of
        Student" + (i+1) + ":");
    System.out.print("Enter name:");
    sc.nextLine();
    String name = sc.nextLine();
    System.out.print("Enter USN:");
    String USN = sc.nextLine();
    System.out.print("Enter semester:");
    int sem = in.nextInt();
    int[] pmarks = new int[5];
    int[] emarks = new int[5];
    System.out.println(); Internals
    System.out.println("Enter marks");
    for(int j=0; j<5; j++){
        System.out.println("Enter Pmarks of"
            + (j+1)+":");
        pmarks[j]=sc.nextInt();
    }
    System.out.println("Enter
        external marks");
    for(int j=0; j<5; j++){
        System.out.println("Enter external
            marks of"+(j+1)+":");
        emarks[j] = sc.nextInt();
    }
    System.out.println();
    stu[i] = new Student(usn, name, sem);
    pm[i] = new Internals(USN, name, sem, pmark
    em[i] = new Externals(USN, name, sem, emarks);
}
System.out.println("Final marks details");
```

```
for (int i=0; i<n; i++){
System.out.println("Student" + (i+1)
                   + ":");
System.out.println("Name" +stu[i].name);
System.out.println("USN:" + stu[i].usn);
System.out.println("Sem:" +stu[i].sem);
for(int j=0; j<5; j++)
{
    fm += im[i].m[j] + em[i].sm[j];
    System.out.println("Final marks of
                course" +(j+1) +":" +fm);
    fm=0;
}
}
}
```

- Enter the no of students
ABC
java.util.InputMismatchException

- Enter the no. of students
2
Enter the details of student 1
Enter the name
XYZ
Enter USN
1BM66
Enter sem
4

Enter the internal marks
Enter the internal marks of 1: 40
Enter the internal mark of 2: 45
Enter the internal marks of 3: 30
Enter the Internal marks of 4: 44
Enter the internal marks of 5: 50

Enter the external marks of 1: 45
Enter the external mark of 2: 50
Enter the external marks of 3: 46
Enter the external mark of 4: 30
Enter the external marks of 5: 43

Final mark obtained
Student 1:
Name: XYZ
USN: 1BM66
Sem: 4
Final marks of course 1: 85
Final marks of course 2: 95
Final marks of course 3: 76
Final marks of course 4: 74
Final marks of course 5: 93

Exception Handling(Father and son)

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

Code:

```
import java.util.*;
class WrongAgeException extends Exception{
        WrongAgeException(String msg){
        System.out.println(msg);
        }
}
class Father{
        int age;
        Father(int age) throws WrongAgeException {
                this.age=age;
                if(age<0){
                throw new WrongAgeException("Age Can't be less than zero!");
                }
                else{
                        System.out.println(" Father's Age Verified!!");
                }
        }
}
class Son extends Father{
        int sonage;
        Son(int age,int sonage)throws WrongAgeException{
        super(age);
        this.sonage=sonage;
        if(sonage<0 || sonage>=age){
                throw new WrongAgeException("Son's age is Invalid!");
        }
        else{
                System.out.println("Son's age verified!");
        }
        }
}
class Age{
        public static void main(String args[]){
                Scanner in=new Scanner(System.in);
                Int age,sonage;
```

```
                System.out.print("Enter Father's Age:");
                age=in.nextInt();
                System.out.print("Enter Son's Age:");
                sonage=in.nextInt();
                try{
                        Son son1=new Son(age,sonage);
                }
                catch(WrongAgeException e1){
                        System.out.println(e1);
                }
                catch (Exception e){
                System.out.println(e);
                }
        }
}
```

Date: / /

Exception handling

Q

ans

```
import java.util.*;
class WrongAgeException extends Exception{
    WrongAgeException(String msg){
        System.out.println(msg);
    }
}

class Father{
    int age;
    Father(int age) throws WrongAgeException{
        this.age=age;
        if(age<0){
            throw new WrongAgeException("Age
                cant be less than zero!");
        }
        else{
            System.out.println("Father's age valid
        }
    }
}
```

```
class Son extends Father {
    int sonage;
    Son(int age, int sonage) throws WrongAgeException {
        super(age);
        this.sonage = sonage;
        if (sonage<0 || sonage >= age) {
            throw new
                WrongAgeException("son's age
                    is invalid");
        }
        else {
            System.out.println("Son's age is
                verified");
        }
    }
}
class Age {
    public static void main(String Args[]) {
        Scanner sl = new Scanner(System.in);
        int age, sonage;
        System.out.println("Enter Father's age:");
        age = sl.nextInt();
        System.out.println("Enter son's age:");
        sonage = sl.nextInt();
        try {
            Son son1 = new Son(age, sonage);
        }
        catch (WrongAgeException e1) {
            System.out.println(e1);
        }
        catch (Exception e) {
            System.out.println(e);
        }
```

}

}

O/P

- Enter Father's age: -10
  Enter Son's age: 12
  Age cant be less than zero!
  WrongAge Exception

- Enter Father's age: 25
  Enter Son's age: -9
  Fathers age verified bb
  Son's age is invalid!
  WrongAge Exception

- Enter Father's age: 35
  Enter Son age: 12
  Fathers age verified bb
  Son's age verified

- Enter Father's age: xyz
  java.util.InputMismatch Exception.

Sagli

Threads

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

Code:

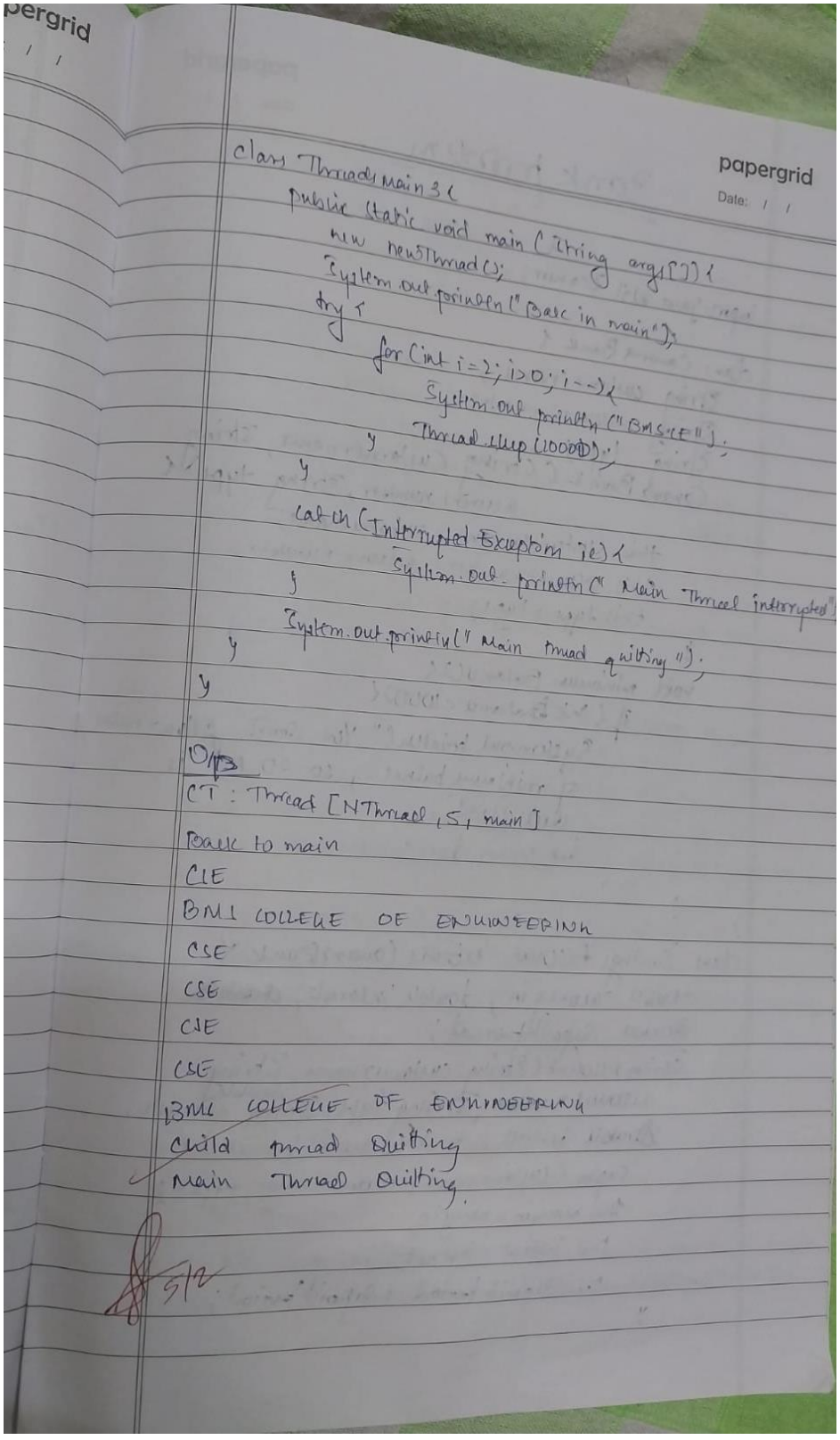
```
class newThread implements Runnable{
	Thread t;
	newThread(){
		t = new Thread(this,"NThread");
		System.out.println("CT: "+t);
		t.start();
	}
	public void run(){
	try{
		for(int i=0;i<5;i++){
			System.out.println("CSE");
			Thread.sleep(100);
		}
	}
	catch(InterruptedException ie){
		System.out.println("Child thread interrupted");
	}
	System.out.println("Child thread quitting");
	}
}
class ThreadsMain3{
	public static void main(String args[]){
		new newThread();
		System.out.println("Back in main");
		try{
			for(int i=0;i<5;i++){
				System.out.println("BMS COLLEGE OF ENGINEERING");
				Thread.sleep(100);
			}
		}
		catch(InterruptedException ie){
			System.out.println("Main Thread Interrupted");
		}
		System.out.println("Main thread quitting");
	}
}
```

```
main : 4
child : 2
main : 3
child 1
child Thread quitting
main : 2
main 1
main thread quitting.
```

Q3

any

```
class newThread implements Runnable
    Thread t;
    newThread(){
        t = new Thread(this, "NThrea")
        System.out.println("CT : "+t);
        t.start();
    }
    public void run(){
        try{
            for(int n=5; n>0; n--){
                System.out.println("Child");
                Thread.sleep(2000);
            }
        }
        Catch(InterruptedException ie){
            System.out.println("Child thread interrupted");
        }
        System.out.println("Child thread Quitting");
    }
```



```
class ThreadMain3 {
    public static void main (String args[]) {
        new newThread();
        System.out.println("Basic in main");
        try {
            for (int i=2; i>0; i--) {
                System.out.println("BMSCE");
                Thread.sleep(10000);
            }
        }
        catch (InterruptedException ie) {
            System.out.println("Main Thread interrupted");
        }
        System.out.println("Main thread exiting");
    }
}
```

O/P

```
CT : Thread[NThread,5,main]
Back to main
CIE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CIE
CSE
BMS COLLEGE OF ENGINEERING
Child thread Quitting
Main Thread Quitting.
```

AWT

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.

Code:

```
import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame implements ActionListener
{
	TextField num1,num2;
	Button dResult;
	Label outResult;
	String out="";
	double resultNum;
	int flag=0;
	public DivisionMain()
	{
		setLayout(new FlowLayout());
		dResult = new Button("RESULT");
		Label number1 = new Label("Number 1:",Label.RIGHT);
		Label number2 = new Label("Number 2:",Label.RIGHT);
		num1=new TextField(5);
		num2=new TextField(5);
		outResult = new Label("Result:",Label.RIGHT);
		add(number1);
		add(num1);
		add(number2);
		add(num2);
		add(dResult);
		add(outResult);
		num1.addActionListener(this);
		num2.addActionListener(this);
		dResult.addActionListener(this);
		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent we){System.exit(0);}
		});
	}
	public void actionPerformed(ActionEvent ae)
	{
		double n1,n2;
		try{
			if (ae.getSource() == dResult){
```

```
                        n1=Double.parseDouble(num1.getText());
                        n2=Double.parseDouble(num2.getText());

                        /*if(n2==0)
                                throw new ArithmeticException();*/
                        out=n1+" "+n2;
                        resultNum=n1/n2;
                        out+=String.valueOf(resultNum);
                        repaint();
                }
        }
        catch(ArithmeticException e2)
        {
                flag=1;
                out="Divide by 0 Exception!  "+e2;
                repaint();
        }
        catch(NumberFormatException e1)
        {
                flag=1;
                out="Number Format Exception!  "+e1;
                repaint();
        }

    }
    public void paint(Graphics g){
        if(flag==0)
                g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+o
                utResult.getHeight()-8);
        else
        g.drawString(out,100,200);
        flag=0;
    }
    public static void main(String[] args){
        DivisionMain dm=new DivisionMain();
        dm.setSize(new Dimension(800,400));
        dm.setTitle("DivionOfIntegers");
        dm.setVisible(true);
    }

}
```

papergrid

Date: / /

AWT

```
import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame implements
ActionListener
{
  TextField num1, num2;
  Button dResult; Label outResult;
  String out=" ";
  double resultNum;
  int flag=0;
  public DivisionMain()
  {
    setLayout(new FlowLayout());
    dResult = new Button("Result");
    Label number1 = new Label("Number:1", Label.RIGHT);
    Label number2 = new Label("Number2:", Label.RIGHT);

    num1 = new TextField(5);
    num2 = new TextField(5);
    outResult = new Label("Result:", Label.RIGHT);
    add(number1);
    add(num1);
    add(number2);
    add(num2);
    add(dResult);
    add(outResult);
    num1.addActionListener(this);
    num2.addActionListener(this);
    dResult.addActionListener(this);
    addWindowListener(new WindowAdapter()
    {
      public void windowClosing(WindowEvent we)
      { System.exit(0); }
    });
  }
```

```
public void actionPerformed (ActionEvent ae)
{
    double n1, n2;
    try{
        if (ae.getSource() == dResult) {
            n1 = Double.parseDouble(num1.getText());
            n2 = Double.parseDouble(num2.getText());
            out = n1 + " " + n2;
            resultNum = n1 / n2
            out += String.valueOf(resultNum);
            repaint();
        }
    }
    catch(ArithmeticException e2)
    {
        flag = 1;
        out = "Divide By 0 Exception" + e2;
        repaint();
    }
    catch(NumberFormatException e1)
    {
        flag = 1;
        out = "Number Format Exception" + e1;
        repaint();
    }
}
public void paint(Graphics g){
    if(flag == 0)
        g.drawString(out, outResult.getX() +
            outResult.getWidth(), outResult.getY()
            + outResult.getHeight() - 8);
    else
        g.drawString(out, 100, 200);
    flag = 0;
```

```
}
}
public static void main(String args[]){
    Division Main dm = new Division Main();
    dm.setSize( new Dimension(800,400));
    dm.setTitle("Division of Integers");
    dm.set Visible(true);
}
}
```

O/P

- Number 1: [12] Number 2: [0]
  [Result] Result: 12 0 0.Infinity
- Number 1: [12] Number 2: [a]
  [Result] Result:

Number FormatExceptions!
java.lang.NumberFormatException: For input string "aaa"

- Number 1: [0] Number 2: [12]
  [Result] Result 0 0 12.00.0
- Number 1: [1] Number 2: [2]
  [Result] Result 1 2 0.5000

21/2