VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Kedar Jevargi (1BM23CS147)

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

Sep-2024 to Jan-2025

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 “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Kedar Jevargi (1BM23CS147),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Surabhi S Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

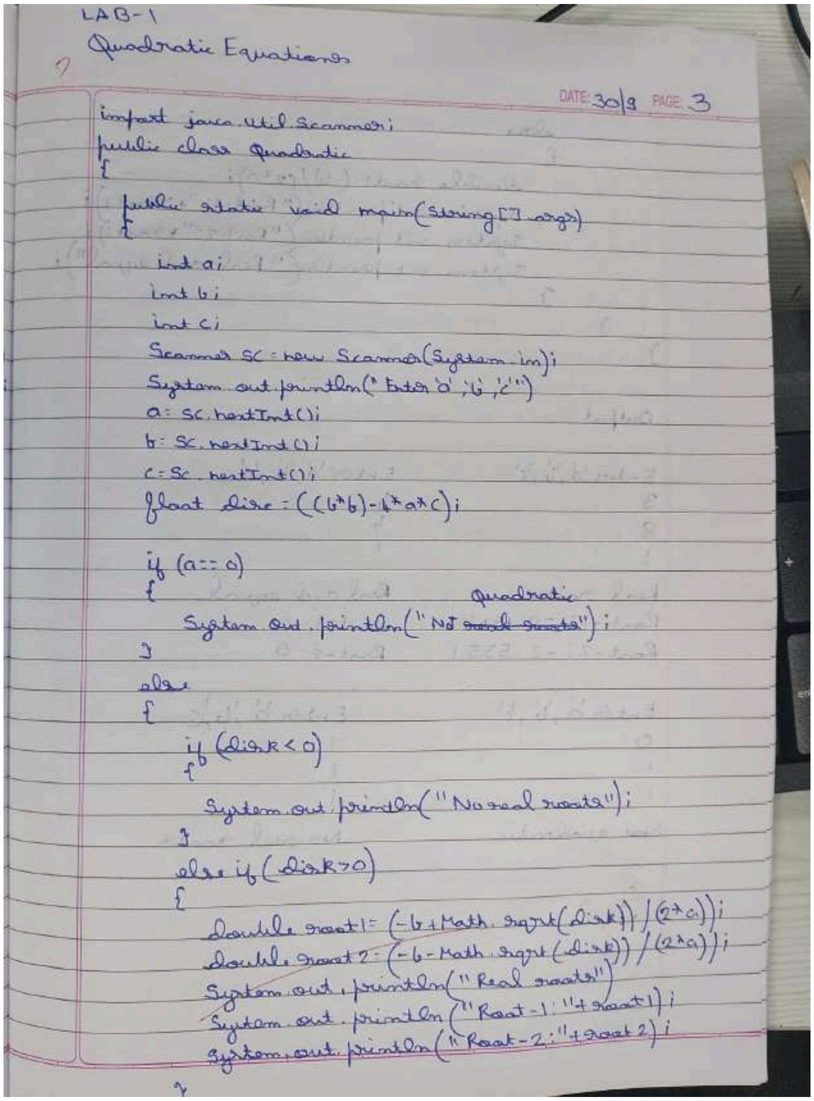
Index

Github Link:
https://github.com/KedarJevargi23/OOJ-CS-LAB-1BM23CS147.git

Program 1

Implement Quadratic Equation

Algorithm:

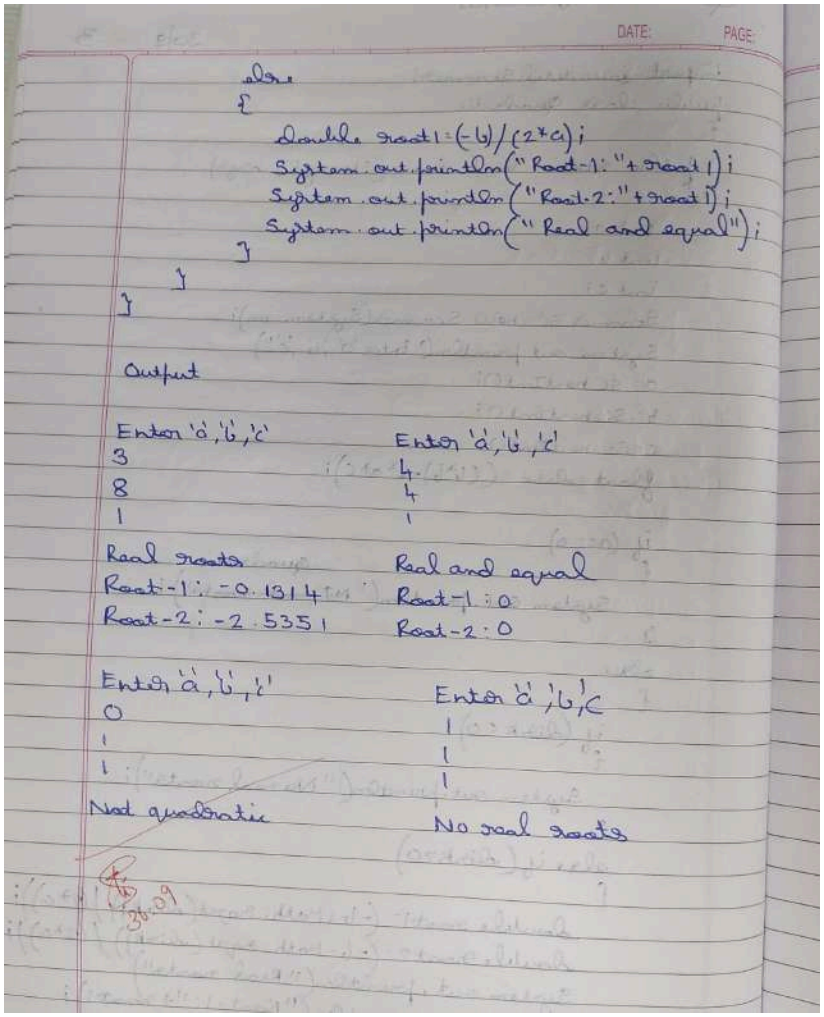


LAB-1

Quadratic Equations

DATE: 30/9 PAGE 3

```
import java.util.Scanner;
public class Quadratic
{
  public static void main(String[] args)
  {
     int a;
     int b;
     int c;
     Scanner sc = new Scanner(System.in);
     System.out.println("Enter 'a','b','c'")
     a= sc.nextInt();
     b= sc.nextInt();
     c= sc.nextInt();
     float disc = ((b*b)-4*a*c);

     if (a== 0)
     {                                    Quadratic
       System.out.println("Not ~~real roots~~");
     }
     else
     {
       if (disk<0)
       {
         System.out.println("No real roots");
       }
       else if (disk>0)
       {
         double root1= (-b+Math.sqrt(disk))/(2*a));
         double root2= (-b-Math.sqrt(disk))/(2*a));
         System.out.println("Real roots")
         System.out.println("Root-1: "+root1);
         System.out.println("Root-2: "+root2);
```



DATE: PAGE:

```
        else
        {
            double root1=(-b)/(2*a);
            System.out.println("Root-1:"+root1);
            System.out.println("Root-2:"+root1);
            System.out.println("Real and equal");
        }
    }
}
```

Output

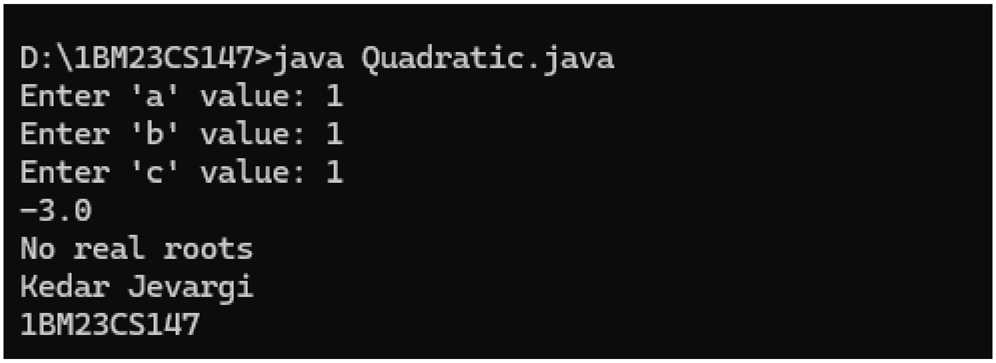
```
Enter 'a','b','c'              Enter 'a','b','c'
3                              4
8                              4
1                              1
Real roots                     Real and equal
Root-1: -0.1314                Root-1: 0
Root-2: -2.5351                Root-2: 0

Enter 'a','b','c'              Enter 'a','b',c
0                              1
1                              1
1                              1
Not quadratic                  No real roots
```

30.09

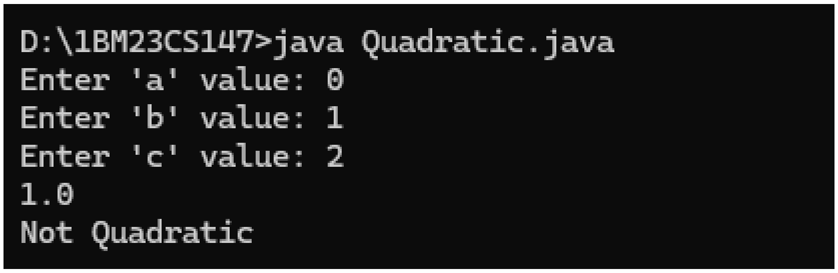
Code:

```
import java.util.Scanner;
public class Quadratic

{
    public static void main(String[] args)
    {
        int a;
        int b;
        int c;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 'a' value: ");
        a= sc.nextInt();
        System.out.print("Enter 'b' value: ");
        b=sc.nextInt();
        System.out.print("Enter 'c' value: ");
        c=sc.nextInt();
        float disc = ((b*b)-4*a*c);
        System.out.println(disc);
        if (a==0)
        {
            System.out.println("Not Quadratic");
        }
        else
        {
            if (disc<0)
            {
            System.out.println("No real roots ");
            }
            else if (disc>0)
            {
            double root1= (-b + Math.sqrt(disc))/(2*a);
            double root2= (-b - Math.sqrt(disc))/(2*a);
            System.out.println("Real roots ");
            System.out.println("Root-1: "+root1);
            System.out.println("Root-2: "+root2);
            }
            else
            {
            double root1=(-b)/(2*a);
                System.out.println("Real and equal");
            System.out.println("Root-1: "+root1);
            System.out.println("Root-2: "+root1);
            }
            System.out.println("Kedar Jevargi");
            System.out.println("1BM23CS147");
```

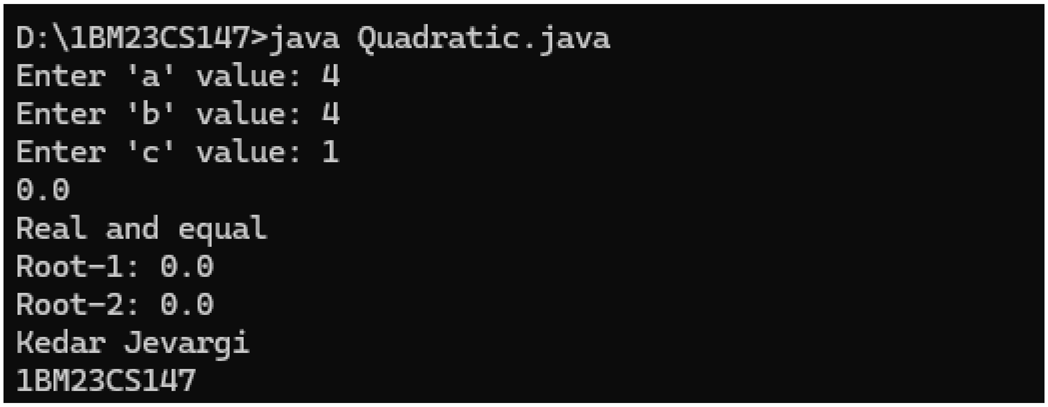
```
      }
   }
}
```



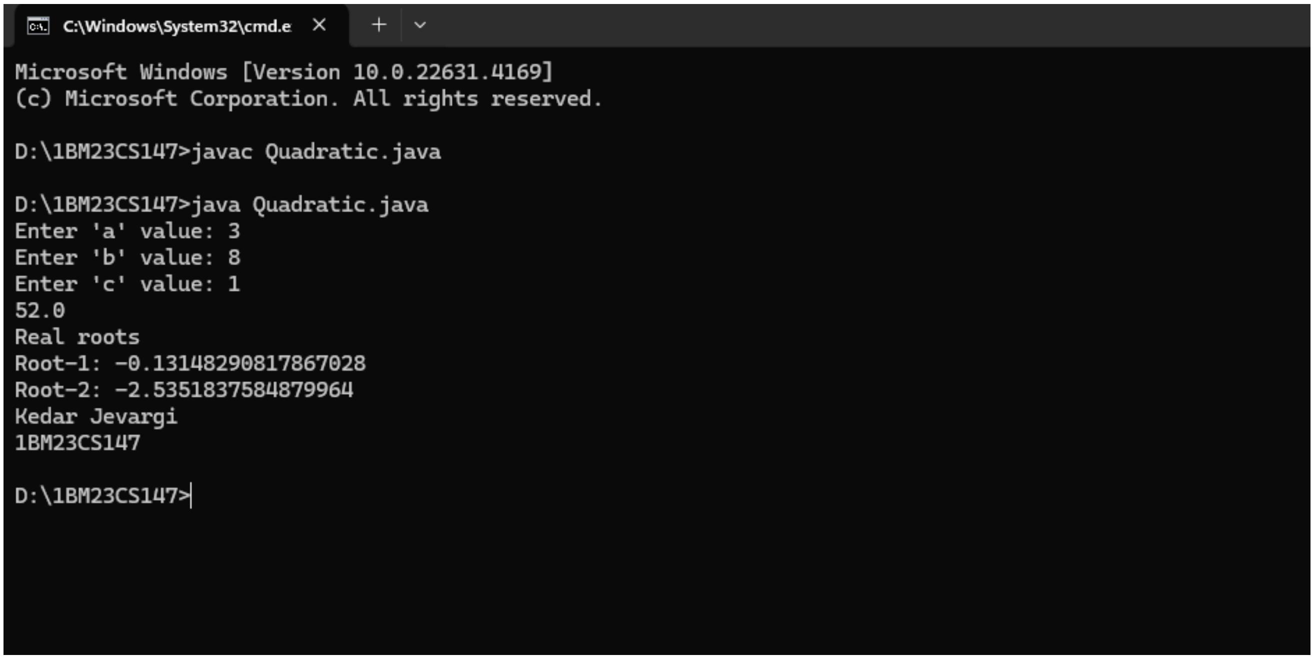
```
D:\1BM23CS147>java Quadratic.java
Enter 'a' value: 1
Enter 'b' value: 1
Enter 'c' value: 1
-3.0
No real roots
Kedar Jevargi
1BM23CS147
```



```
D:\1BM23CS147>java Quadratic.java
Enter 'a' value: 0
Enter 'b' value: 1
Enter 'c' value: 2
1.0
Not Quadratic
```



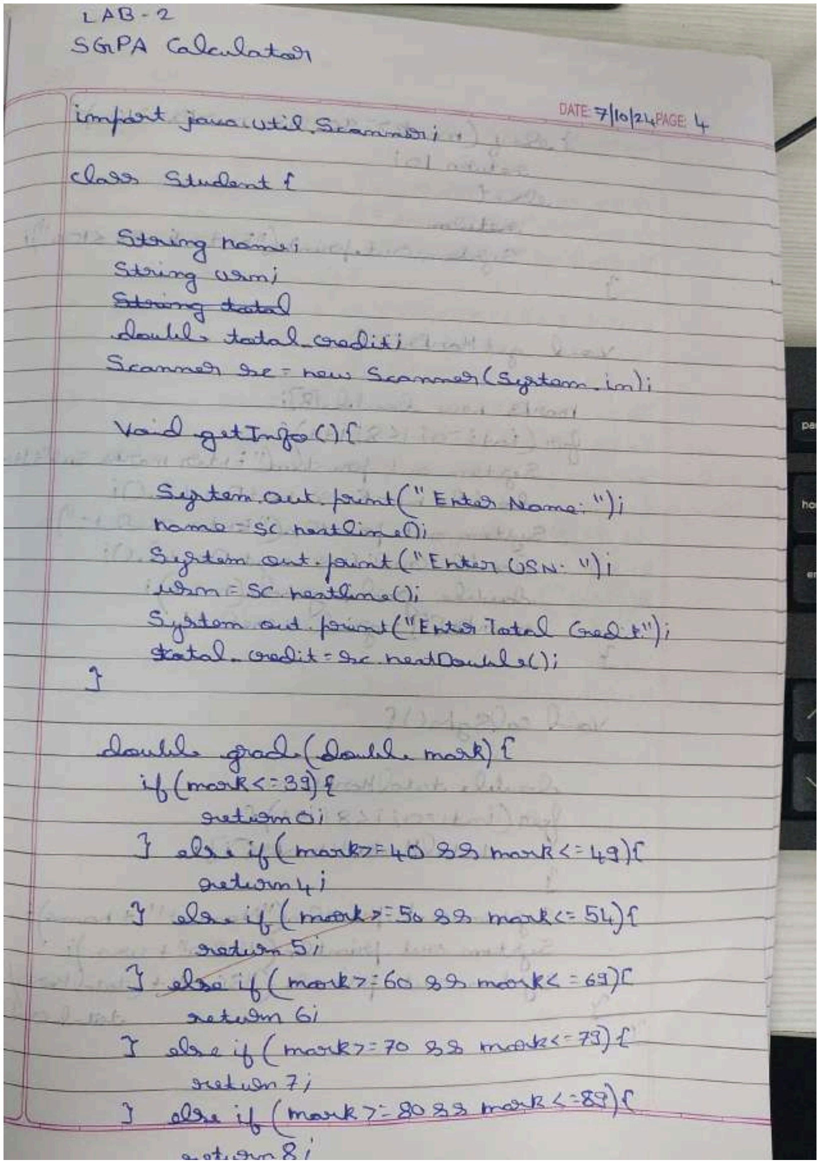
```
D:\1BM23CS147>java Quadratic.java
Enter 'a' value: 4
Enter 'b' value: 4
Enter 'c' value: 1
0.0
Real and equal
Root-1: 0.0
Root-2: 0.0
Kedar Jevargi
1BM23CS147
```



```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS147>javac Quadratic.java

D:\1BM23CS147>java Quadratic.java
Enter 'a' value: 3
Enter 'b' value: 8
Enter 'c' value: 1
52.0
Real roots
Root-1: -0.13148290817867028
Root-2: -2.5351837584879964
Kedar Jevargi
1BM23CS147

D:\1BM23CS147>
```

Program 2
SGPA Calculator

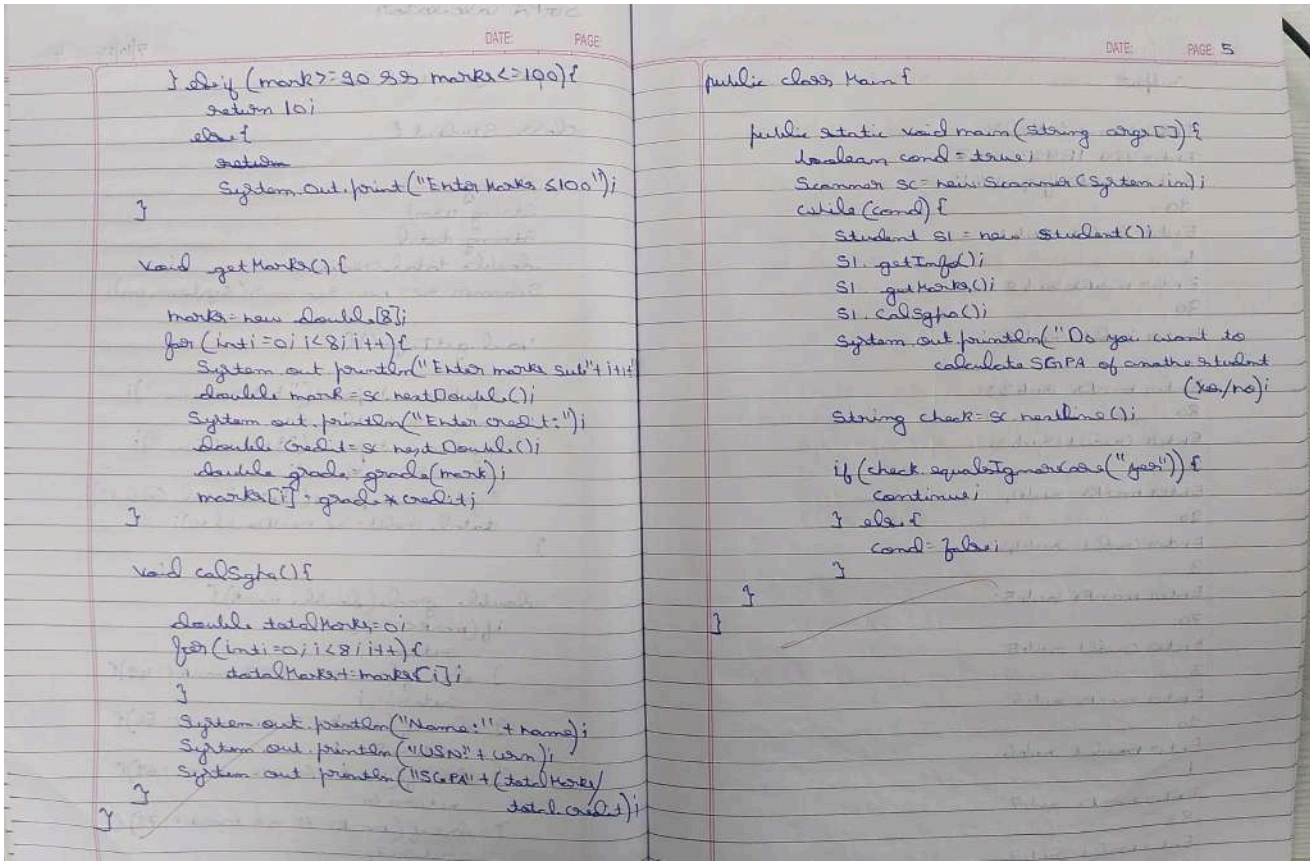
Algorithm:



LAB - 2
SGPA Calculator

DATE: 7/10/24 PAGE: 4

```
import java.util.Scanner;

class Student {

    String name;
    String usn;
    ~~String total~~
    double total_credit;
    Scanner sc = new Scanner(System.in);

    Void getInfo(){

        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter USN: ");
        usn = SC.nextLine();
        System.out.print("Enter Total Credit");
        total_credit = sc.nextDouble();
   }

   double grad(double mark){
      if(mark<=39){
          return 0;
      } else if(mark>=40 && mark<=49){
          return 4;
      } else if(mark>=50 && mark<=54){
          return 5;
      } else if(mark>=60 && mark<=69){
          return 6;
      } else if(mark>=70 && mark<=79){
          return 7;
      } else if(mark>=80 && mark<=89){
          return 8;
```



DATE: PAGE:

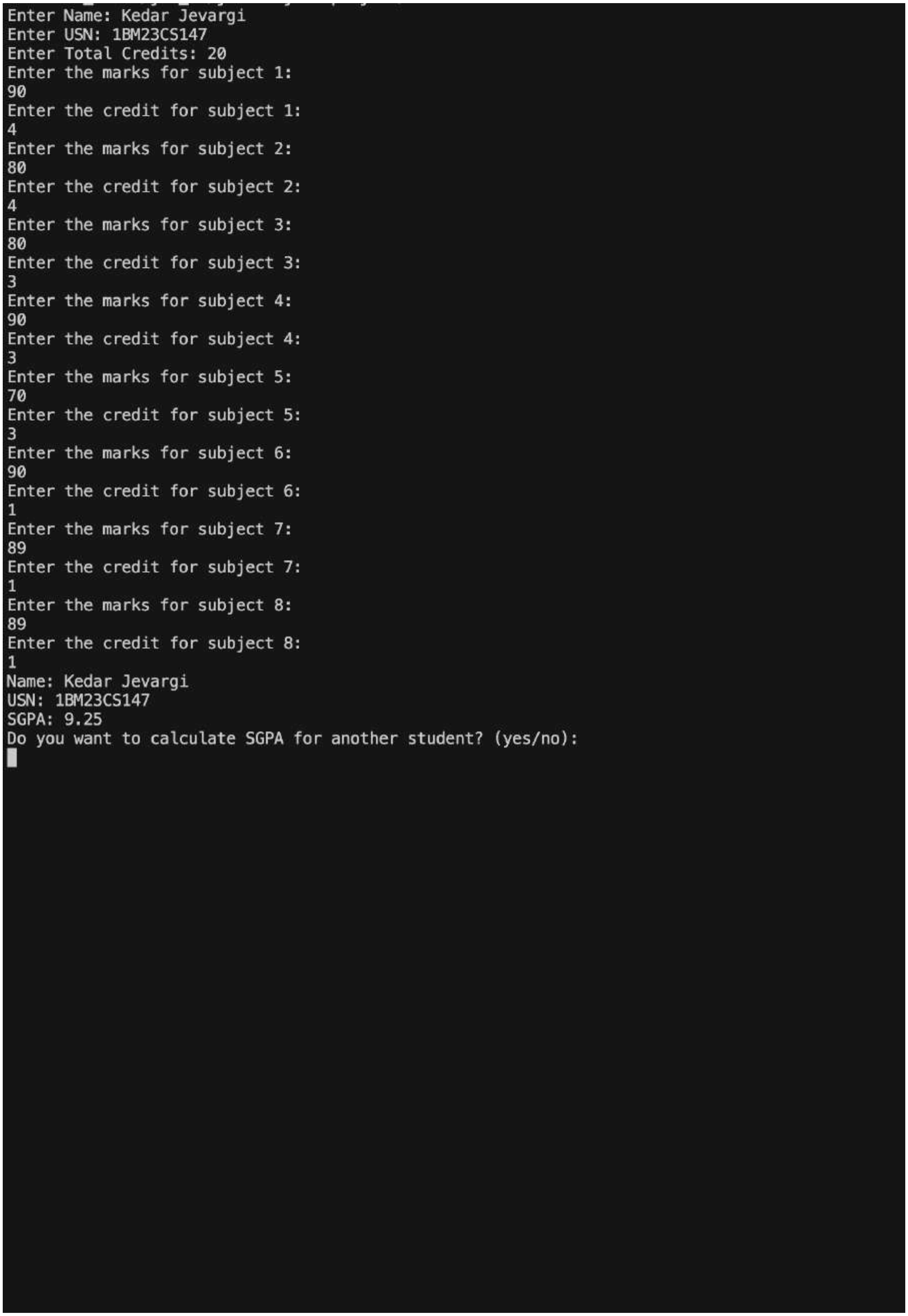
```
    } else if (mark>=90 && marks<=100){
        return 10;
    else{
        ~~return~~
        System.out.print("Enter Marks ≤100");
}

void getMarks(){

    marks=new Double[8];
    for(int i=0;i<8;i++){
        System.out.println("Enter marks sub"+i+1);
        double mark=sc.nextDouble();
        System.out.println("Enter credit:");
        Double Credit=sc.nextDouble();
        double grade=grade(mark);
        marks[i]=grade*credit;
}

void calSgpa(){

    double totalMarks=0;
    for(int i=0;i<8;i++){
        totalMarks+=marks[i];
    }
    System.out.println("Name:"+name);
    System.out.println("USN"+usn);
    System.out.println("SGPA"+(totalMarks/
                                   total.credit);
  }
}
```

DATE: PAGE: 5

```
public class Main{

    public static void main(String args[]){
        boolean cond=true;
        Scanner sc=new Scanner(System.in);
        while(cond){
            Student S1=new Student();
            S1.getInfo();
            S1.getMarks();
            S1.calSgpa();
            System.out.println("Do you want to
                    calculate SGPA of another student
                                          (yes/no);
            String check=sc.nextLine();

            if(check.equalsIgnoreCase("yes")){
                continue;
            } else{
                cond=false;
            }
    }
}
```

Code:

```
import java.util.Scanner;

class Student {
    private String name;
    private String usn;
    private double total_credit;
    private double[] marks;
    private Scanner sc = new Scanner(System.in);

    void getInfo() {
        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter USN: ");
        usn = sc.nextLine();

        System.out.print("Enter Total Credits: ");
        total_credit = sc.nextDouble();
        sc.nextLine();
    }

    double grade(double mark) {
        if (mark <= 39) {
            return 0;
        } else if (mark >= 40 && mark <= 49) {
            return 4;
        } else if (mark >= 50 && mark <= 54) {
            return 5;
        } else if (mark >= 55 && mark <= 59) {
            return 6;
        } else if (mark >= 60 && mark <= 69) {
            return 7;
        } else if (mark >= 70 && mark <= 79) {
            return 8;
        } else if (mark >= 80 && mark <= 89) {
            return 9;
        } else {
            return 10;
        }
    }

    void getMarks() {
        marks = new double[8];
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter the marks for subject " + (i + 1) + ": ");
            double mark = sc.nextDouble();
            System.out.println("Enter the credit for subject " + (i + 1) + ": ");
            double credit = sc.nextDouble();

            double grade = grade(mark);
            marks[i] = grade * credit;
        }
        sc.nextLine();
    }

    void calSgpa() {
        double totalMarks = 0;
        for (int i = 0; i < 8; i++) {
            totalMarks += marks[i];
        }
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + (totalMarks / total_credit));
    }
}

public class Main {
    public static void main(String args[]) {
        boolean cond = true;
        Scanner sc = new Scanner(System.in);
        while (cond) {
            Student s1 = new Student();


            s1.getInfo();
            s1.getMarks();
            s1.calSgpa();

            System.out.println("Do you want to calculate SGPA for another student? (yes/no): ");
            String check = sc.nextLine();
            if (check.equalsIgnoreCase("yes")) {
                continue;
            } else {
                cond = false;
            }
        }
        sc.close();
    }
}
```

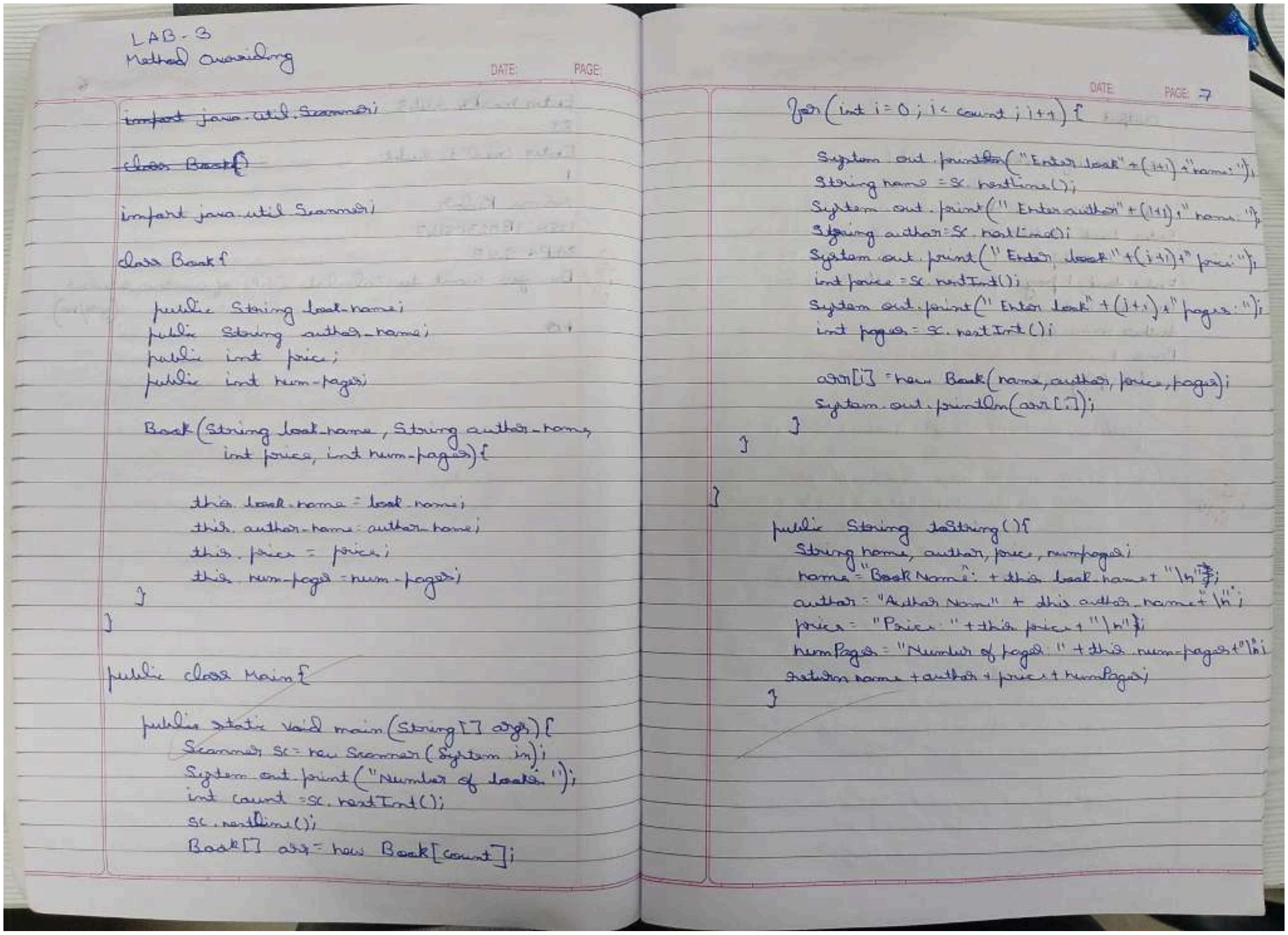


```
Enter Name: Kedar Jevargi
Enter USN: 1BM23CS147
Enter Total Credits: 20
Enter the marks for subject 1:
90
Enter the credit for subject 1:
4
Enter the marks for subject 2:
80
Enter the credit for subject 2:
4
Enter the marks for subject 3:
80
Enter the credit for subject 3:
3
Enter the marks for subject 4:
90
Enter the credit for subject 4:
3
Enter the marks for subject 5:
70
Enter the credit for subject 5:
3
Enter the marks for subject 6:
90
Enter the credit for subject 6:
1
Enter the marks for subject 7:
89
Enter the credit for subject 7:
1
Enter the marks for subject 8:
89
Enter the credit for subject 8:
1
Name: Kedar Jevargi
USN: 1BM23CS147
SGPA: 9.25
Do you want to calculate SGPA for another student? (yes/no):
```

Program 3

Method Overriding

Algorithm:



LAB-3
Method overriding

DATE: PAGE:

```
~~import java.util.Scanner;~~

~~class Book{~~

import java.util.Scanner;

class Book{

    public String book-name;
    public String author_name;
    public int price;
    public int num-pages;

    Book(String book-name, String author-name,
           int price, int num-pages){

        this.book.name = book-name;
        this.author-name = author-name;
        this.price = price;
        this.num-pages = num-pages;
    }
}

public class Main{

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Number of books: ");
        int count = sc.nextInt();
        sc.nextLine();
        Book[] arr = new Book[count];
```

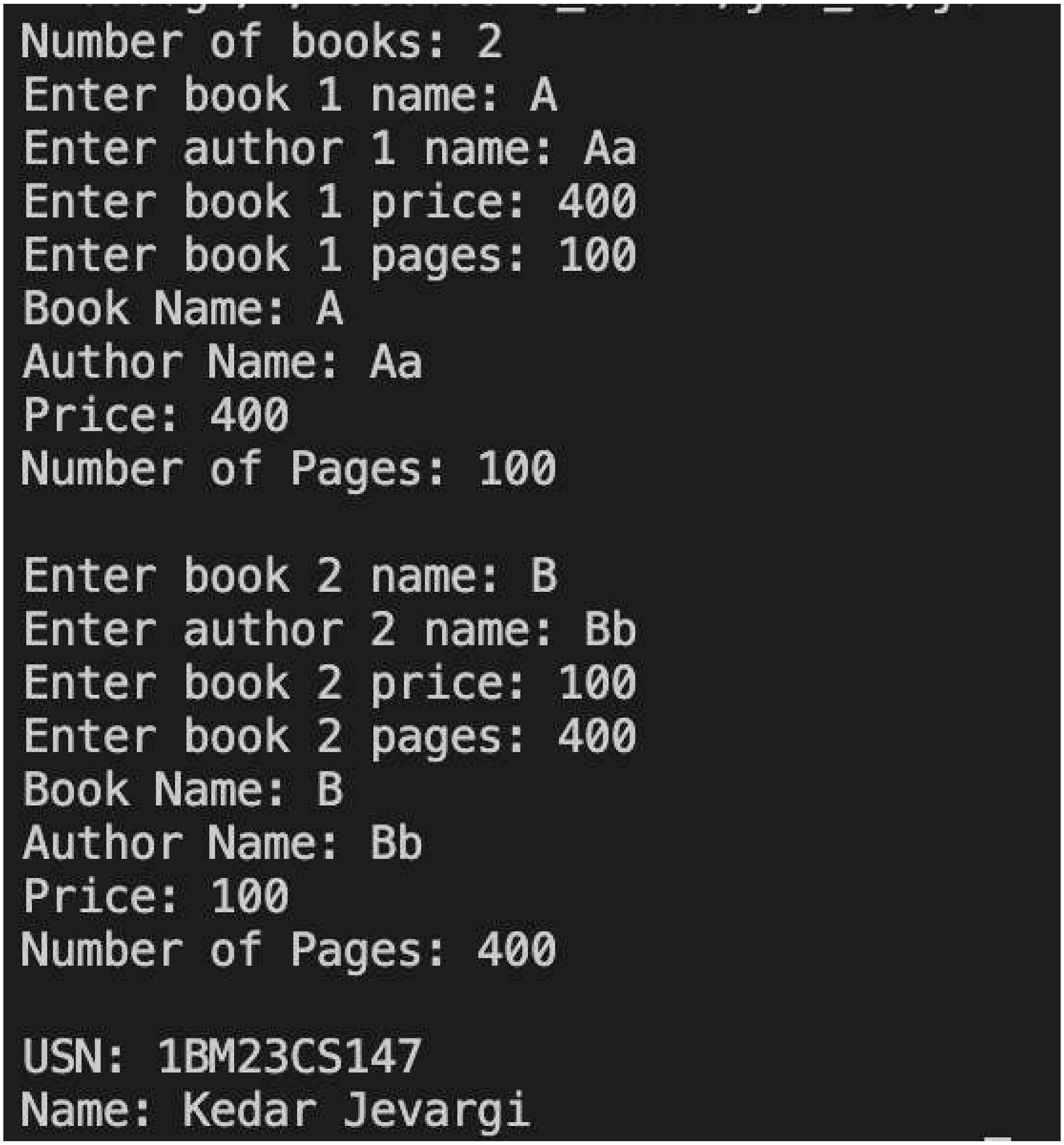
DATE: PAGE: 7

```
    for(int i = 0; i < count; i++){

        System.out.println("Enter book" + (i+1) + "name: ");
        String name = sc.nextLine();
        System.out.print(" Enter author" + (i+1) + " name: ");
        String author = sc.nextLine();
        System.out.print(" Enter book" + (i+1) + " price: ");
        int price = sc.nextInt();
        System.out.print(" Enter book" + (i+1) + " pages: ");
        int pages = sc.nextInt();

        arr[i] = new Book(name, author, price, pages);
        System.out.println(arr[i]);
    }
  }

}

  public String toString(){
    String name, author, price, numpages;
    name = "Book Name: " + this.book_name + "\n";
    author = "Author Name" + this.author_name + "\n";
    price = "Price: " + this.price + "\n";
    numPages = "Number of pages: " + this.num-pages + "\n";
    return name + author + price + numPages;
  }
```

Code:

```
import java.util.Scanner;

class Book {
   public String book_name;
   public String author_name;
   public int price;
   public int num_pages;

   Book(String book_name, String author_name, int price, int num_pages) {
      this.book_name = book_name;
      this.author_name = author_name;
      this.price = price;
      this.num_pages = num_pages;
   }

   @Override
   public String toString() {
      String name, author, price, numPages;
      name = "Book Name: " + this.book_name + "\n";
      author = "Author Name: " + this.author_name + "\n";
      price = "Price: " + this.price + "\n";
      numPages = "Number of Pages: " + this.num_pages + "\n";
      return name + author + price + numPages;
   }
}

public class ride {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Number of books: ");
      int count = sc.nextInt();
      sc.nextLine();

      Book[] arr = new Book[count];
      for (int i = 0; i < count; i++) {
         System.out.print("Enter book " + (i + 1) + " name: ");
         String name = sc.nextLine();
         System.out.print("Enter author " + (i + 1) + " name: ");
         String author = sc.nextLine();
         System.out.print("Enter book " + (i + 1) + " price: ");
         int price = sc.nextInt();
         System.out.print("Enter book " + (i + 1) + " pages: ");
         int pages = sc.nextInt();
         sc.nextLine();

         arr[i] = new Book(name, author, price, pages);
```

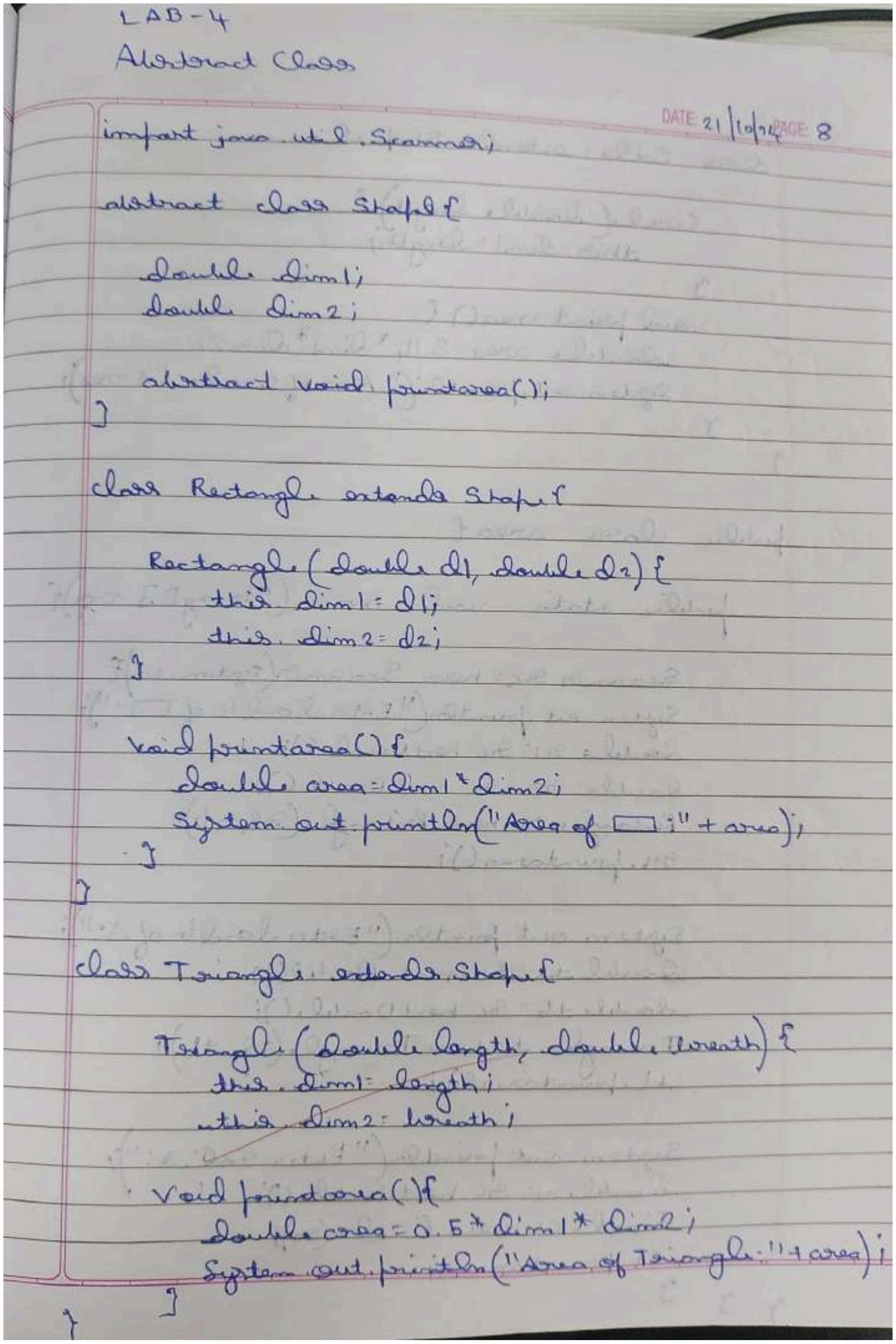
```
            System.out.println(arr[i]);
        }

        sc.close();

        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
    }
}
```



```
Number of books: 2
Enter book 1 name: A
Enter author 1 name: Aa
Enter book 1 price: 400
Enter book 1 pages: 100
Book Name: A
Author Name: Aa
Price: 400
Number of Pages: 100

Enter book 2 name: B
Enter author 2 name: Bb
Enter book 2 price: 100
Enter book 2 pages: 400
Book Name: B
Author Name: Bb
Price: 100
Number of Pages: 400

USN: 1BM23CS147
Name: Kedar Jevargi
```

Program 4
Abstract Class

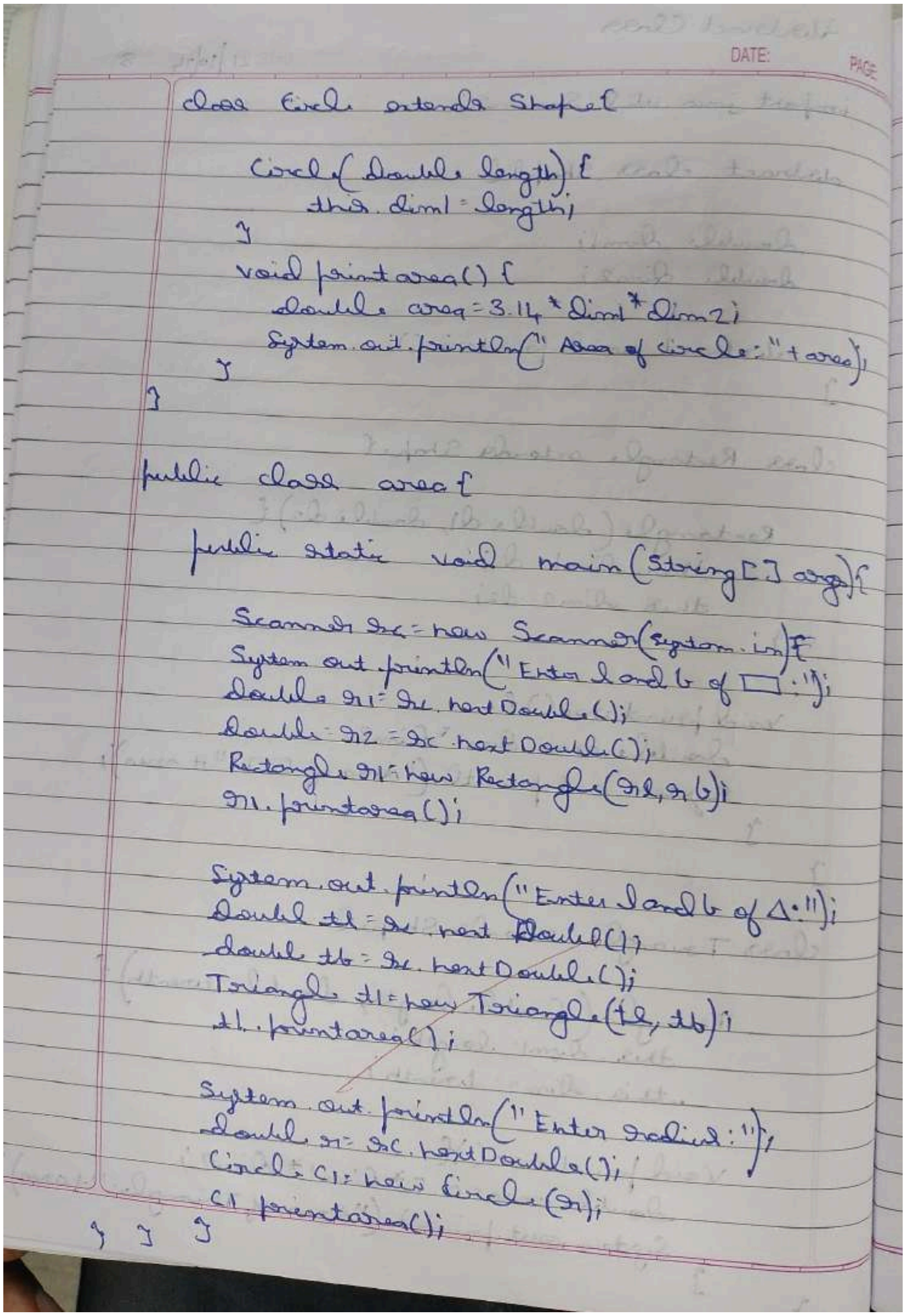
Algorithm:



LAB-4
Abstract Class

DATE 21/10/24 PAGE 8

```
import java.util.Scanner;

abstract class Shape{

  double dim1;
  double dim2;

  abstract void printarea();
}

class Rectangle extends Shape{

   Rectangle(double d1, double d2){
       this.dim1=d1;
       this.dim2=d2;
   }

   void printarea(){
       double area=dim1*dim2;
       System.out.println("Area of ▭:"+area);
   }
}

class Triangle extends Shape{

   Triangle(double length, double breath){
       this.dim1=length;
       this.dim2=breath;

   void printarea(){
       double area=0.5*dim1*dim2;
       System.out.println("Area of Triangle:"+area);
   }
}
```

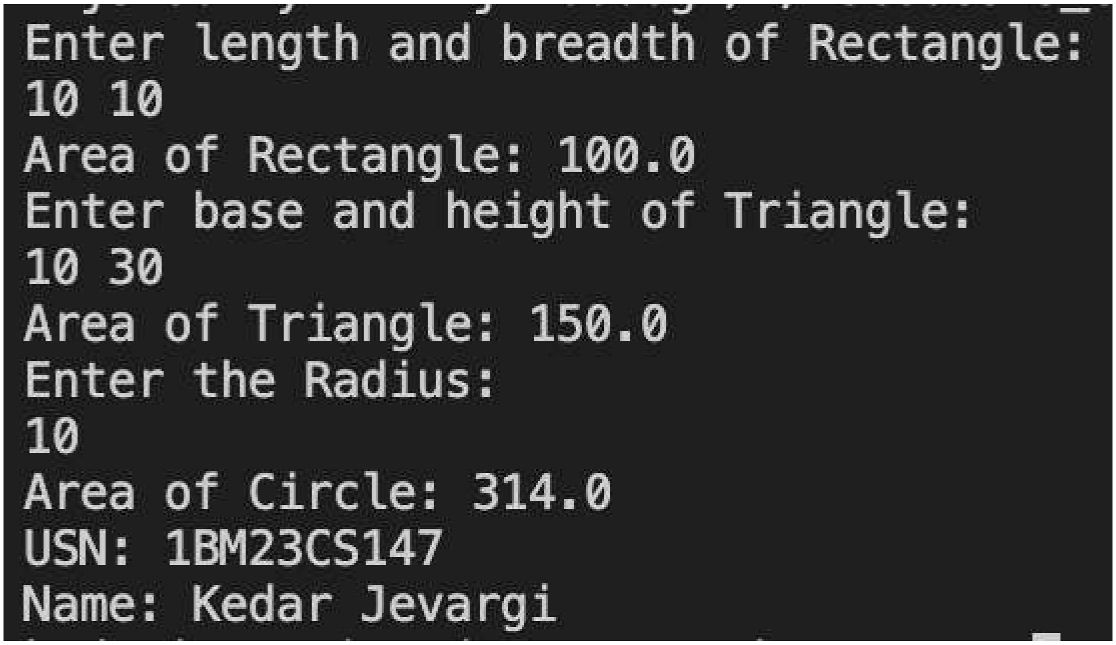


```
class Circle extends Shape{

    Circle(Double length){
        this.dim1 = length;
    }
    void printarea(){
        double area=3.14*dim1*dim2;
        System.out.println(" Area of circle:"+area);
    }
}

public class area{

    public static void main(String[] args){

        Scanner sc= new Scanner(system.in);
        System out println("Enter l and b of □:");
        Double r1= sc.nextDouble();
        Double r2 =sc nextDouble();
        Rectangle r1= new Rectangle(rl,rb);
        r1.printarea();

        System.out.println("Enter l and b of Δ:");
        Doubl tl = sc.next Double();
        double tb = sc.nextDouble();
        Triangle t1= new Triangle(tl, tb);
        t1.printarea();

        System.out.println("Enter radius:");
        double r= sc.nextDouble();
        Circle c1= new Circle(r);
        c1.printarea();
} } }
```

Code:

```
import java.util.Scanner;

abstract class Shape {
   double dim1;
   double dim2;

   abstract void printarea();
}

class Rectangle extends Shape {
   Rectangle(double d1, double d2) {
      this.dim1 = d1;
      this.dim2 = d2;
   }

   @Override
   void printarea() {
      double area = dim1 * dim2;
      System.out.println("Area of Rectangle: " + area);
   }
}

class Triangle extends Shape {
   Triangle(double base, double height) {
      this.dim1 = base;
      this.dim2 = height;
   }

   @Override
   void printarea() {
      double area = 0.5 * dim1 * dim2;
      System.out.println("Area of Triangle: " + area);
   }
}

class Circle extends Shape {
   Circle(double radius) {
      this.dim1 = radius;
   }

   @Override
   void printarea() {
      double area = 3.14 * dim1 * dim1;
      System.out.println("Area of Circle: " + area);
   }
}
```

```
public class area {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter length and breadth of Rectangle:");
            double rl = sc.nextDouble();
            double rb = sc.nextDouble();
            Rectangle r1 = new Rectangle(rl, rb);
            r1.printarea();

            System.out.println("Enter base and height of Triangle:");
            double base = sc.nextDouble();
            double height = sc.nextDouble();
            Triangle t1 = new Triangle(base, height);
            t1.printarea();

            System.out.println("Enter the Radius:");
            double radius = sc.nextDouble();
            Circle c1 = new Circle(radius);
            c1.printarea();
        }

        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
    }
}
```

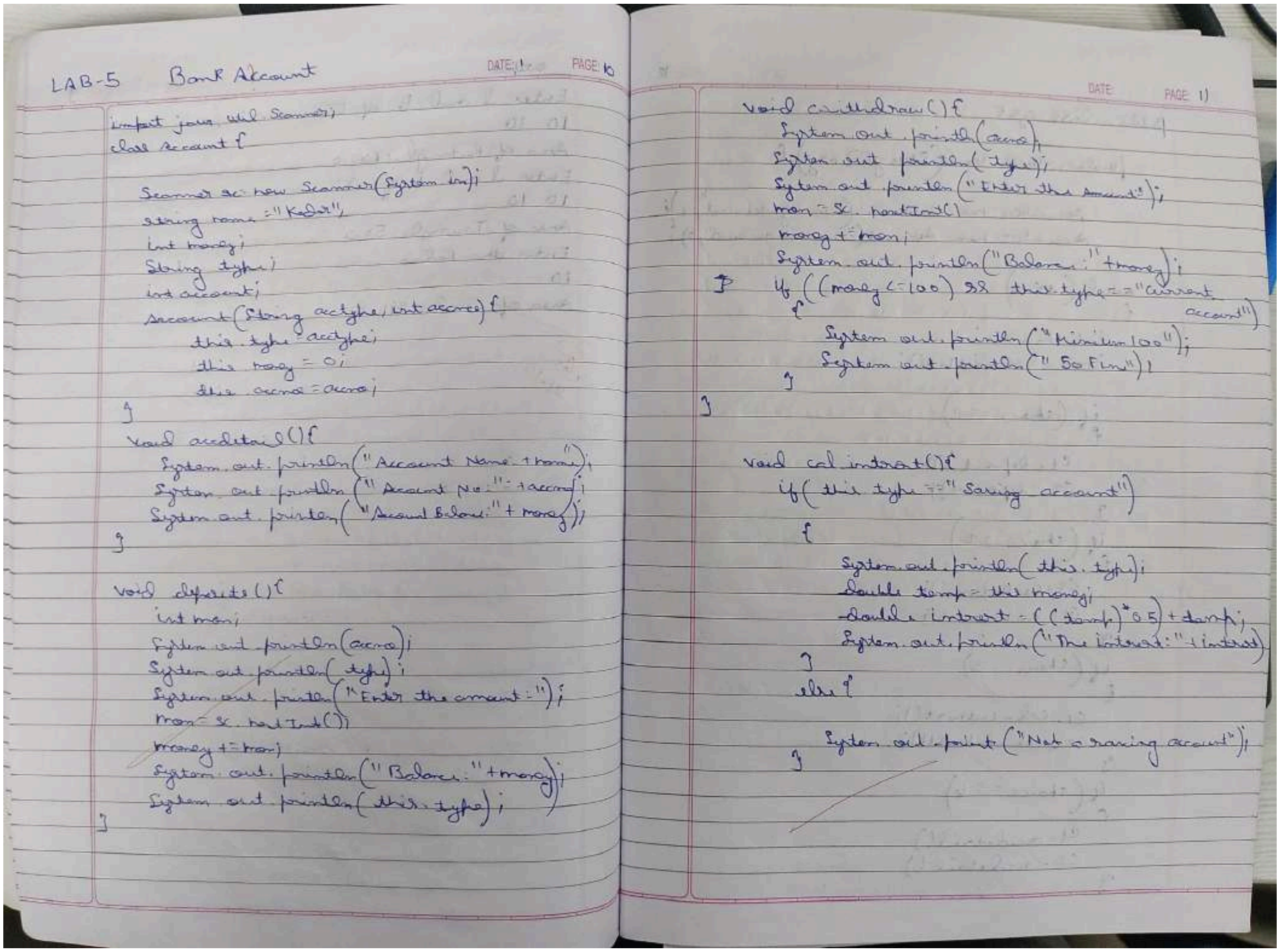


```
Enter length and breadth of Rectangle:
10 10
Area of Rectangle: 100.0
Enter base and height of Triangle:
10 30
Area of Triangle: 150.0
Enter the Radius:
10
Area of Circle: 314.0
USN: 1BM23CS147
Name: Kedar Jevargi
```

Program 5

Bank Account

Algorithm:



LAB-5 Bank Account

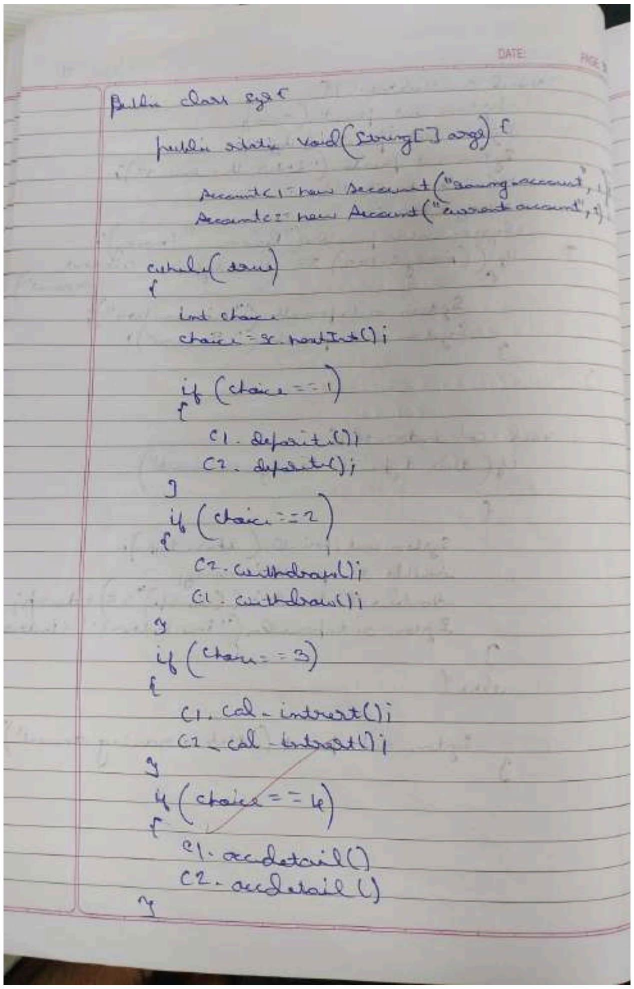
DATE: PAGE: 10

```
import java.util.Scanner;
class Account {

    Scanner sc= new Scanner(System.in);
    string name ="Kedar";
    int money;
    String type;
    int account;
    Account(String acctype, int accno){
        this.type=acctype;
        this.money = 0;
        this.accno =accno;
    }
    void accdetail(){
        System.out.println("Account Name: +name");
        System.out.println("Account No:"+accno);
        System.out.println("Account Balance:"+ money);
}

void deposite(){
    int mon;
    System.out.println(accno);
    System.out.println(type);
    System.out.println("Enter the amount:");
    mon= sc.nextInt();
    money += mon;
    System.out.println("Balance:"+money);
    System.out.println(this.type);
}
```

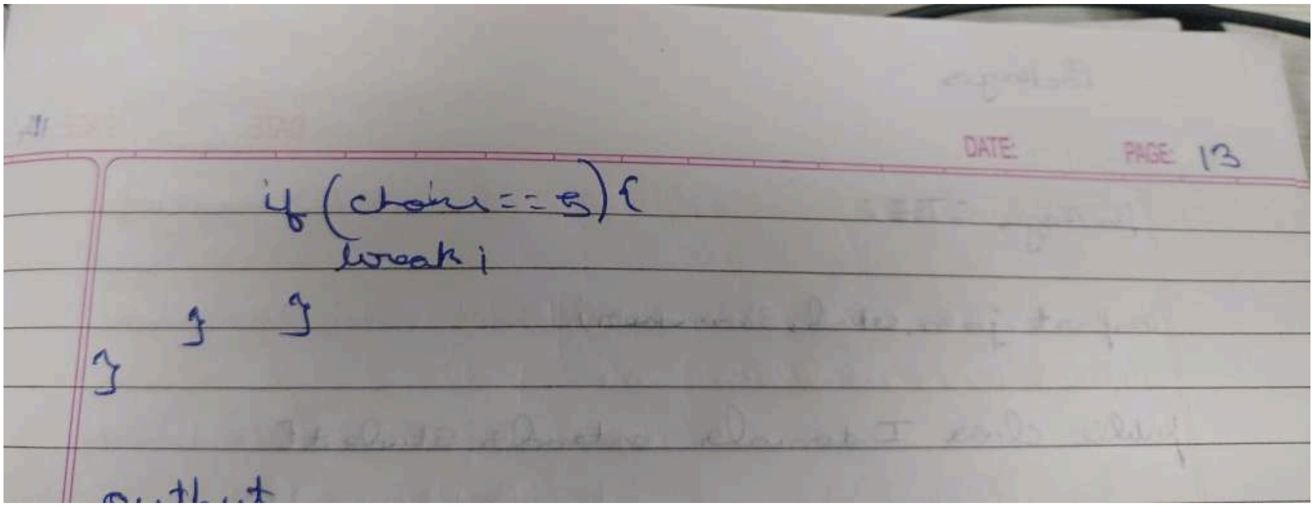
DATE PAGE 11

```
void withdraw(){
    System.out.println(accno);
    System.out.println(type);
    System.out.println("Enter the amount:");
    mon= sc.nextInt()
    money += mon;
    System.out.println("Balance:"+money);
    if((money <=100) && this.type=="current account")
    {
        System.out.println("Minimum 100");
        System.out.println(" So Fine");
    }
}

void cal interest(){
    if(this.type=="Saving account")
    {
        System.out.println(this.type);
        Double temp= this.money;
        Double interest =((temp)*0.5)+temp;
        System.out.println("The interest:"+interest)
    }
    else{

        System.out.print("Not a saving account");
    }
```

21



```
Public class Syz{
    public static void(String[] args) {

        Account c1= new Account("saving account", 1
        Account c2= new Account("current account", 2)

    while(true)
    {
        int choice
        choice = sc.nextInt();

        if (choice == 1)
        {
            c1. deposit();
            c2. deposit();
        }
        if (choice == 2)
        {
            c2. withdraw();
            c1. withdraw();
        }
        if (choice == 3)
        {
            c1. cal_intrest();
            c2_cal_intrest();
        }
        if (choice == 4)
        {
            c1. accdetail()
            c2. accdetail()
        }
```



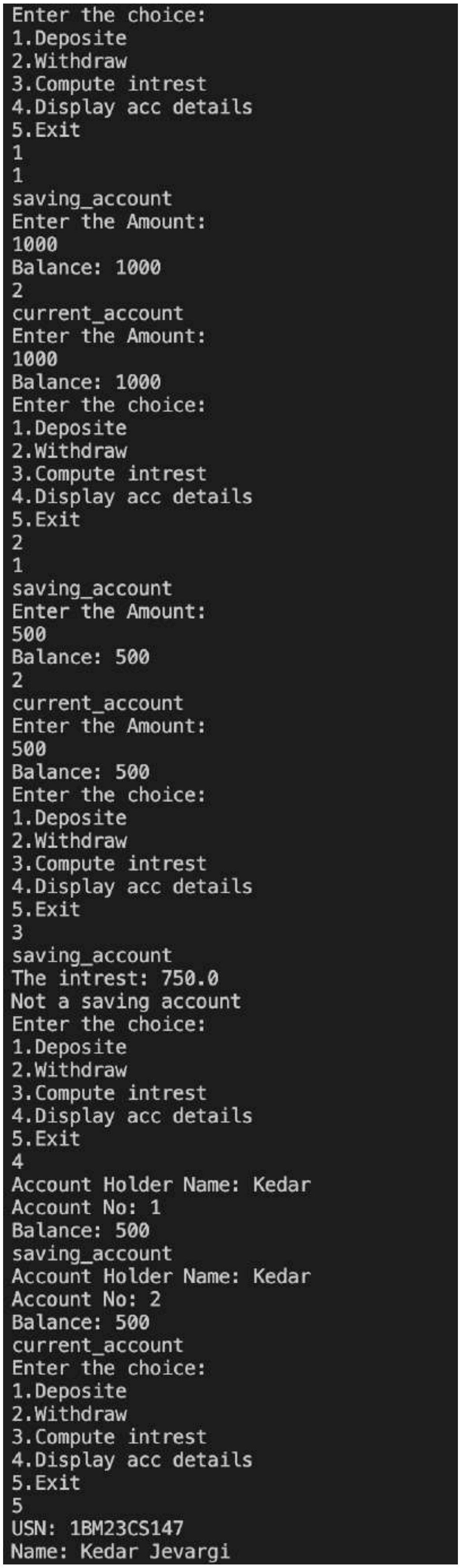
DATE:
PAGE: 13

```
            if (choice == 5){
                break;

        }   }
}
```

Output

Code:

```
import java.util.Scanner;

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

  String name="Kedar";
  int money;
  String type;
  int accno;
  Account(String acctype,int accno){
    this.type=acctype;
    this.money=0;
    this.accno=accno;

  }
  void accdetail(){
    System.out.println("Account Holder Name: "+name);
    System.out.println("Account No: "+accno);
    System.out.println("Balance: "+money);
    System.out.println(this.type);
  }

  void deposit(){
    int mon;
    System.out.println(accno);
    System.out.println(type);
    System.out.println("Enter the Amount: ");
    mon=sc.nextInt();
    money+=mon;
    System.out.println("Balance: "+money);
  }

  void withdraw(){
    System.out.println(this.accno);
    System.out.println(type);
    int mon;
    System.out.println("Enter the Amount: ");
    mon=sc.nextInt();
    money-=mon;
    System.out.println("Balance: "+money);
    if((money<=100) && this.type=="current_account")
    {
      System.out.println("Minimum balance is 100");
      System.out.println("Deposite money now and pay the fine of 50");
    }
  }
  void cal_intrest(){
    if(this.type=="saving_account")
    {

      System.out.println(this.type);
      double temp=this.money;

      double intrest=((temp)*0.5)+temp;
      System.out.println("The intrest: "+intrest);
    }

    else
    {
      System.out.println("Not a saving account");
    }

  }
}

public class sys {
    public static void main(String[] args) {
      Account c1=new Account("saving_account",1);
      Account c2=new Account("current_account",2);

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

        int choice;
        System.out.println("Enter the choice:\n1.Deposite\n2.Withdraw\n3.Compute intrest\n4.Display acc details\n5.Exit");
        choice=sc.nextInt();
        if (choice==1)
        {
          c1.deposit();
          c2.deposit();
        }

        if(choice==2){
           c1.withdraw();
           c2.withdraw();
        }

        if(choice==3){
          c1.cal_intrest();
          c2.cal_intrest();
       }
       if(choice==4){
        c1.accdetail();
        c2.accdetail();
     }

     if(choice==5){
      break;
      }
    }

    // Adding USN and Name
    System.out.println("USN: 1BM23CS147");
    System.out.println("Name: Kedar Jevargi");
  }
}
```

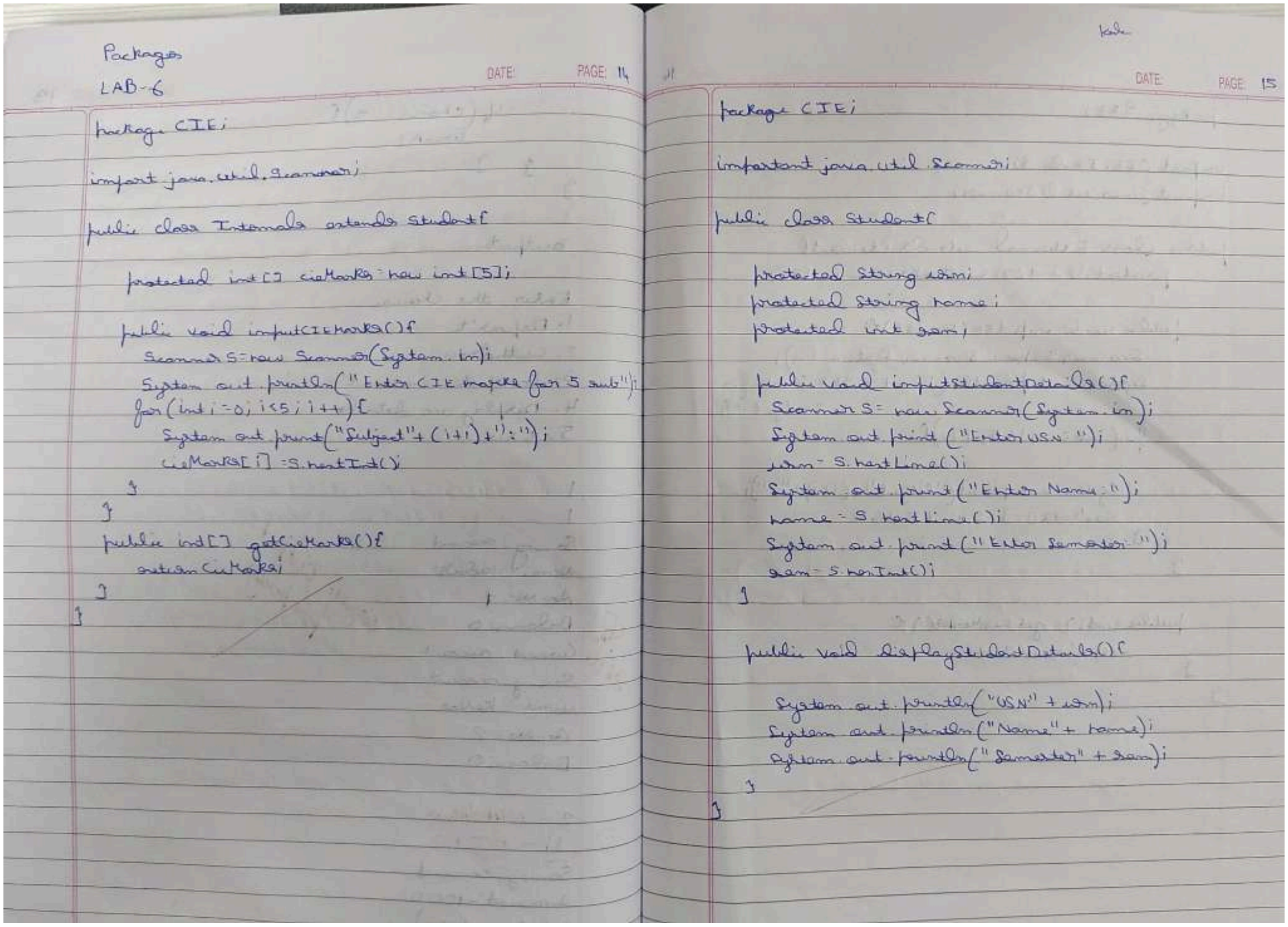


```
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
1
1
saving_account
Enter the Amount:
1000
Balance: 1000
2
current_account
Enter the Amount:
1000
Balance: 1000
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
2
1
saving_account
Enter the Amount:
500
Balance: 500
2
current_account
Enter the Amount:
500
Balance: 500
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
3
saving_account
The intrest: 750.0
Not a saving account
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
4
Account Holder Name: Kedar
Account No: 1
Balance: 500
saving_account
Account Holder Name: Kedar
Account No: 2
Balance: 500
current_account
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
5
USN: 1BM23CS147
Name: Kedar Jevargi
```

Program 6

Packages

Algorithm:



Packages

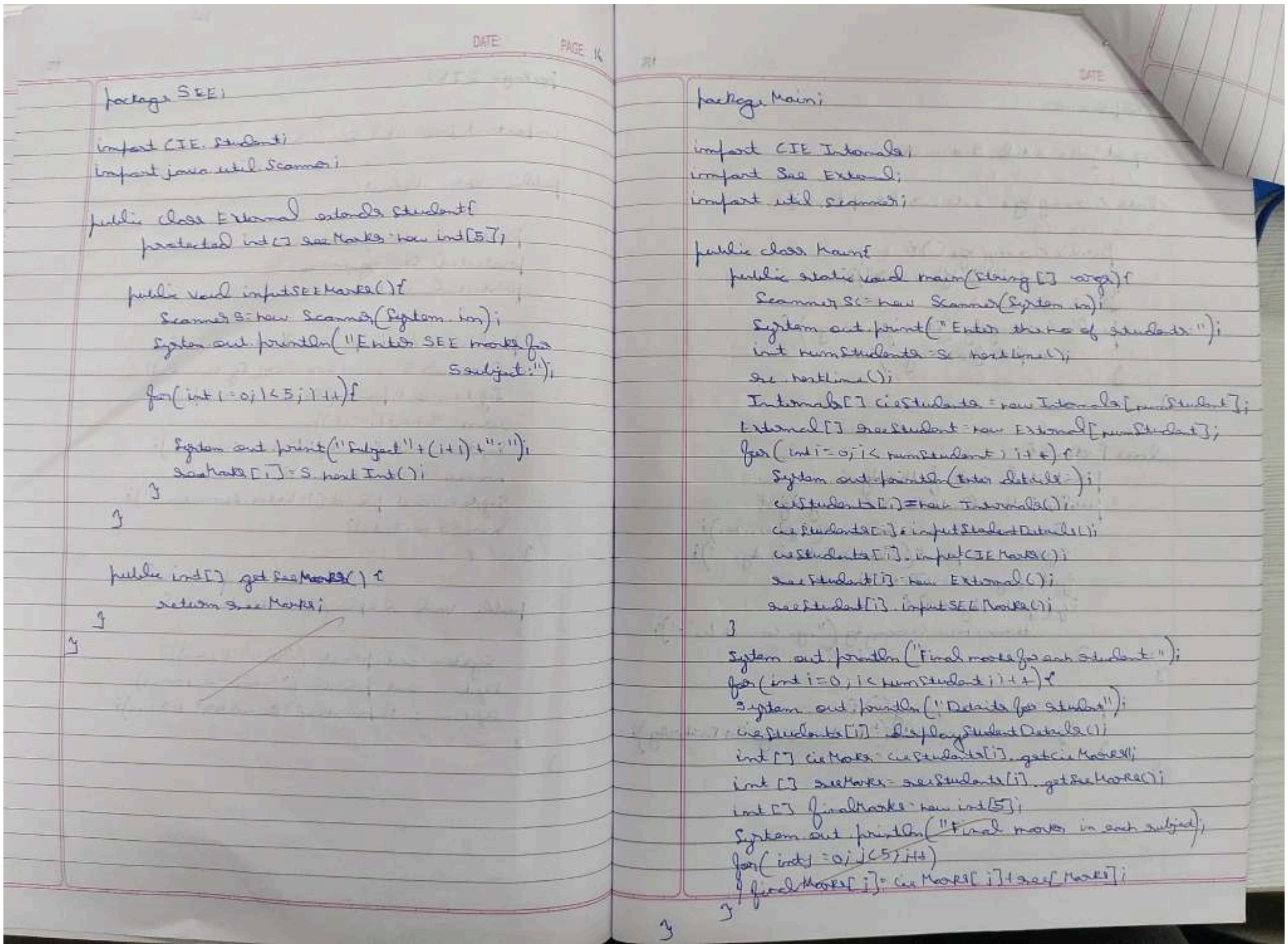
LAB-6

DATE: PAGE: 14

```
package CIE;

import java.util.Scanner;

public class Internals extends Student{

    protected int[] cieMarks=new int[5];

    public void inputCIEMarks(){
      Scanner S=new Scanner(System.in);
      System.out.println("Enter CIE marks for 5 sub");
      for(int i=0; i<5; i++){
        System.out.print("Subject"+(i+1)+":");
        cieMarks[i]=S.nextInt();
      }
   }
  public int[] getCieMarks(){
     return CieMarks;
  }
}
```

DATE: PAGE: 15

```
package CIE;

important java.util.Scanner;

public class Student{

   protected String usn;
   protected String name;
   protected int sem;

   public void inputStudentDetails(){
     Scanner S= new Scanner(System.in);
     System.out.print("Enter USN: ");
     usn= S.nextLine();
     System.out.print("Enter Name: ");
     name= S.nextLine();
     System.out.print("Enter Semester: ");
     sem= S.nextInt();
  }

  public void displayStudentDetails(){

     System.out.println("USN" + usn);
     System.out.println("Name"+ name);
     System.out.println("Semester" + sem);
  }
}
```



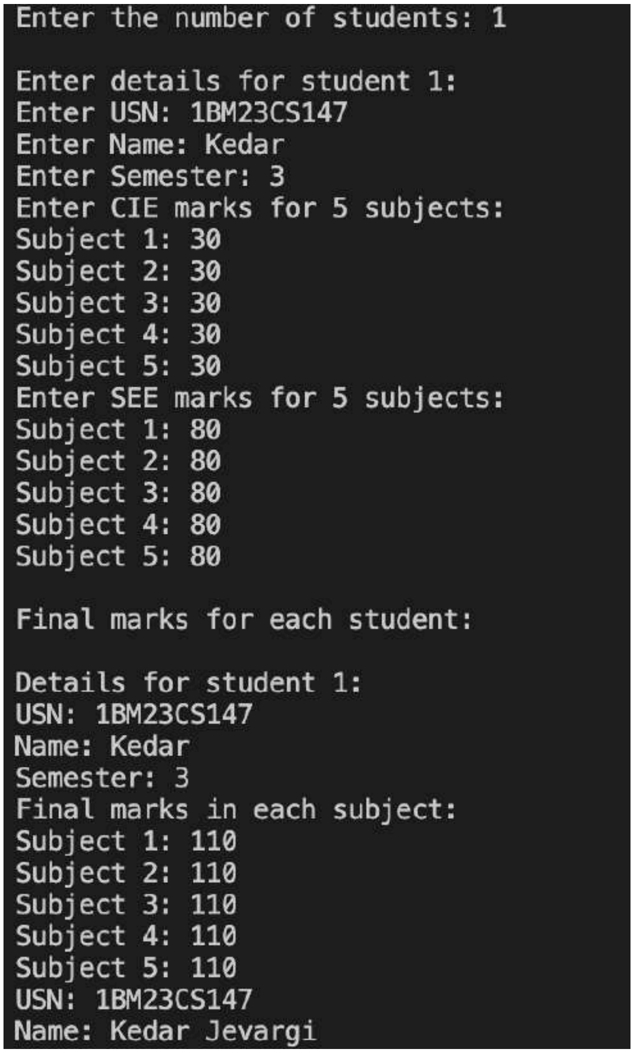
```
package SEE;

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

public class External extends Student{
      protected int[] seeMarks=new int[5];

   public void inputSEEMarks(){
      Scanner S=new Scanner(System.in);
      System.out.println("Enter SEE marks for
                                      5 subject:");
      for(int i=0;i<5;i++){

         System.out.print("Subject"+(i+1)+":");
         seeMarks[i]=S.nextInt();
      }
   }

   public int[] getSeeMarks() {
         return seeMarks;
   }
}
```

```
package Main;

import CIE.Internals;
import See.External;
import util.Scanner;

public class Main{
   public static void main(String[] args){
      Scanner SC=new Scanner(System.in);
      System.out.print("Enter the no of students:");
      int numStudents=Sc.nextLine();
      Sc.nextLine();
      Internals[] cieStudents=new Internals[numStudent];
     External[] seeStudent=new External[numStudent];
     for(int i=0;i<numStudent;i++){
        System.out.println(Enter details:);
        cieStudents[i]=new Internals();
        cieStudents[i].inputStudentDetails();
        cieStudents[i].inputCIEMarks();
        seeStudent[i]=new External();
        seeStudent[i].inputSEEMarks();
     }
    System.out.println("Final marks for each student:");
    for(int i=0;i<numStudent;i++){
     System.out.println("Details for student");
     cieStudents[i].displayStudentDetails();
     int[] cieMarks=cieStudents[i].getcieMarks();
     int[] seeMarks=seeStudents[i].getSeeMarks();
     int[] finalMarks=new int[5];
     System.out.println("Final marks in each subject");
     for(int j=0;j<5;j++)
     {finalMarks[j]=cieMarks[j]+see[Marks];
    }
  }
}
```

Code:

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int[] cieMarks = new int[5];

    public void inputCIEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            cieMarks[i] = s.nextInt();
        }
    }

    public int[] getCieMarks() {
        return cieMarks;
    }
}


package CIE;

import java.util.Scanner;

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

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
```

```
}

package Main;

import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

        System.out.print("Enter the number of students: ");
        int numStudents = sc.nextInt();
        sc.nextLine();

        Internals[] cieStudents = new Internals[numStudents];
        External[] seeStudents = new External[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");

            cieStudents[i] = new Internals();
            cieStudents[i].inputStudentDetails();
            cieStudents[i].inputCIEMarks();

            seeStudents[i] = new External();
            seeStudents[i].inputSEEMarks();
        }

        System.out.println("\nFinal marks for each student:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nDetails for student " + (i + 1) + ":");
            cieStudents[i].displayStudentDetails();

            int[] cieMarks = cieStudents[i].getCieMarks();
            int[] seeMarks = seeStudents[i].getSeeMarks();
            int[] finalMarks = new int[5];

            System.out.println("Final marks in each subject:");
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = cieMarks[j] + seeMarks[j];
                System.out.println("Subject " + (j + 1) + ": " + finalMarks[j]);
            }
        }
        sc.close();

        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
    }
}


package SEE;

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

public class External extends Student {
    protected int[] seeMarks = new int[5];

    public void inputSEEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

    public int[] getSeeMarks() {
        return seeMarks;
    }
}
```

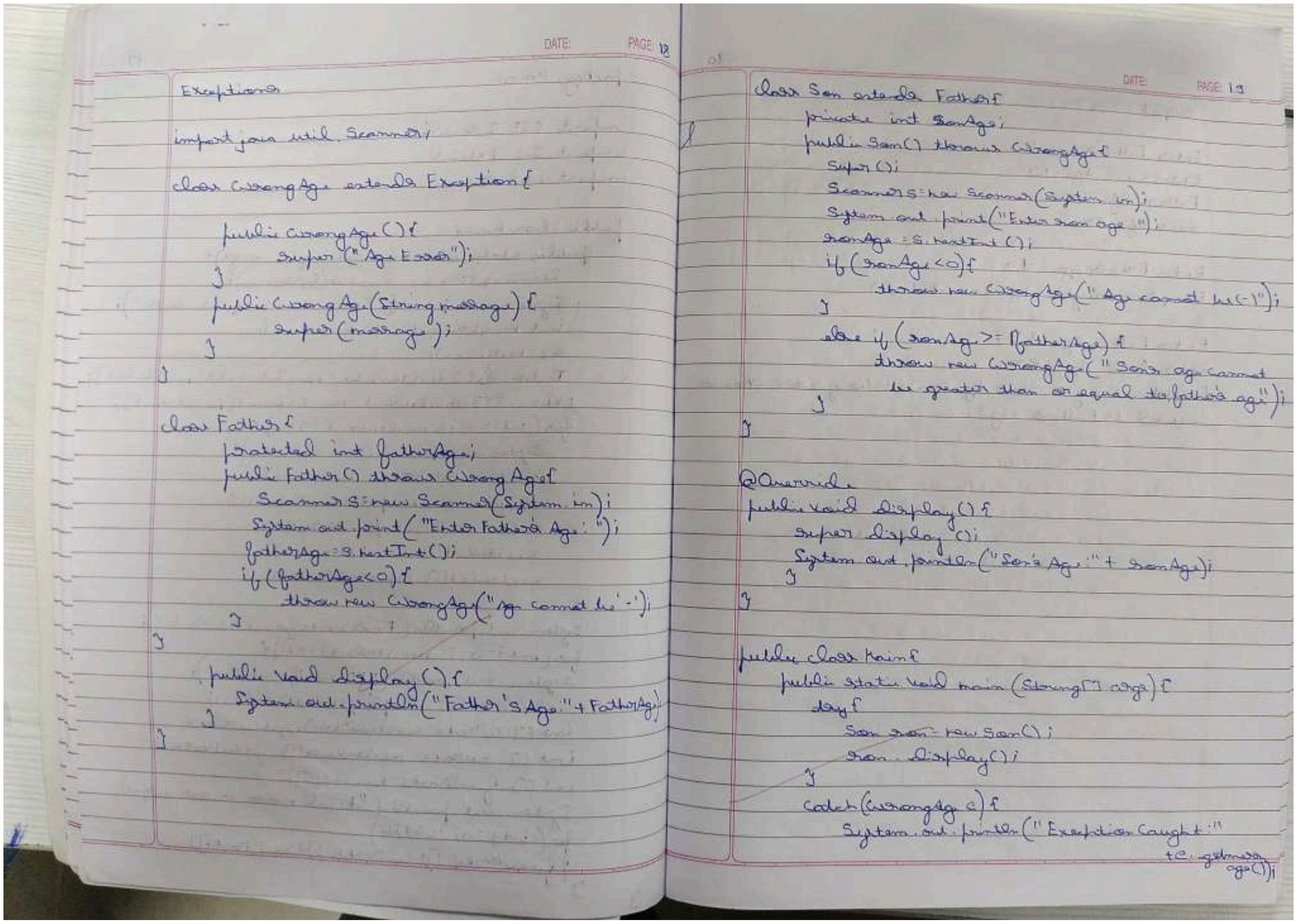


```
Enter the number of students: 1

Enter details for student 1:
Enter USN: 1BM23CS147
Enter Name: Kedar
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject 1: 30
Subject 2: 30
Subject 3: 30
Subject 4: 30
Subject 5: 30
Enter SEE marks for 5 subjects:
Subject 1: 80
Subject 2: 80
Subject 3: 80
Subject 4: 80
Subject 5: 80

Final marks for each student:

Details for student 1:
USN: 1BM23CS147
Name: Kedar
Semester: 3
Final marks in each subject:
Subject 1: 110
Subject 2: 110
Subject 3: 110
Subject 4: 110
Subject 5: 110
USN: 1BM23CS147
Name: Kedar Jevargi
```

Program 7

Exception Handling

Algorithm:



DATE: PAGE: 18

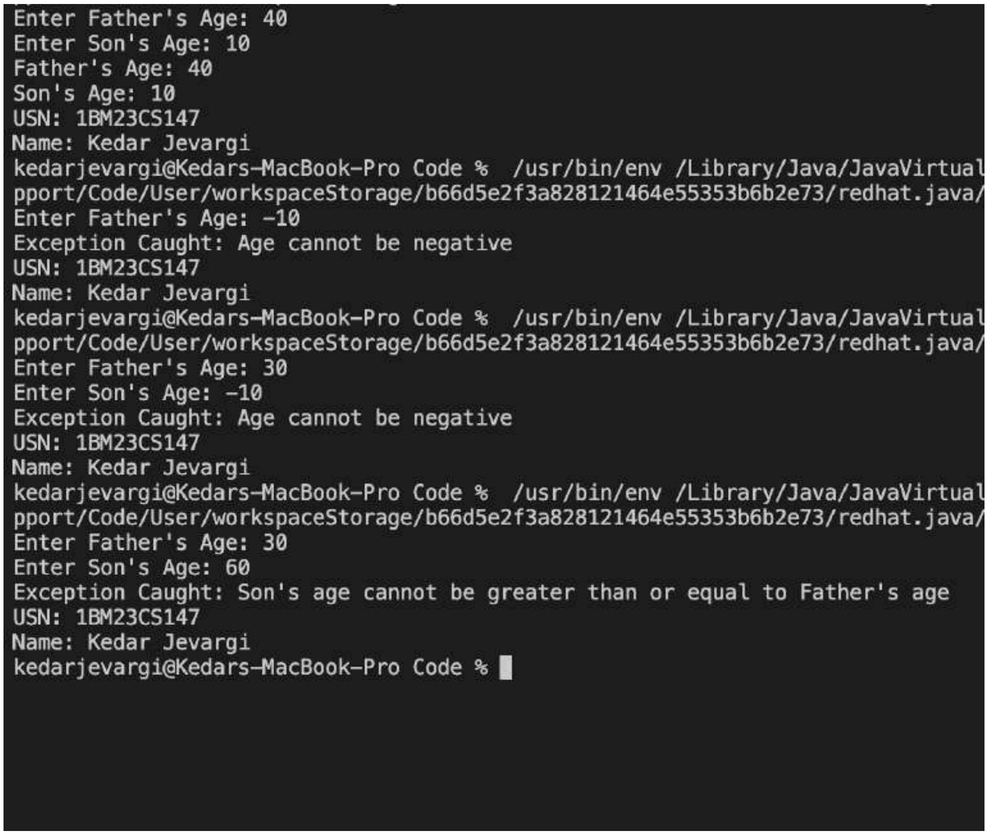
```
Exceptions

import java.util.Scanner;

class WrongAge extends Exception{

    public WrongAge(){
        super("Age Error");
    }
    public WrongAge(String message) {
        super(message);
    }
}

class Father{
    protected int fatherAge;
    public Father() throws WrongAge{
        Scanner S=new Scanner(System.in);
        System.out.print( "Enter Father's Age: ");
        fatherAge=S.nextInt();
        if (fatherAge<0){
            throw new WrongAge("Age cannot be '-');
        }
}

    public void display(){
        System.out.println("Father's Age:"+FatherAge);
    }
}
```

DATE: PAGE: 19

```
class Son extends Father{
    private int sonAge;
    public Son() throws WrongAge{
        Super();
        Scanner S=new Scanner(System.in);
        System.out.print("Enter son age: ");
        sonAge=S.nextInt();
        if(sonAge<0){
            throw new WrongAge("Age cannot be(-)");
        }
        else if (sonAge>=fatherAge) {
            throw new WrongAge("Son's age cannot
                be greater than or equal to father's age!");
        }
}

@Override
public void display(){
    super.display();
    System.out.println("Son's Age:"+ sonAge);
    }
}

public class Main{
    public static void main (String[] args){
        try{
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println("Exception Caught:"
                                    +e.getMessage());
```

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's Age: ");
        sonAge = s.nextInt();
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
        }
    }

    @Override
    public void display() {
        super.display();
        System.out.println("Son's Age: " + sonAge);
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }

        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
    }
}
```

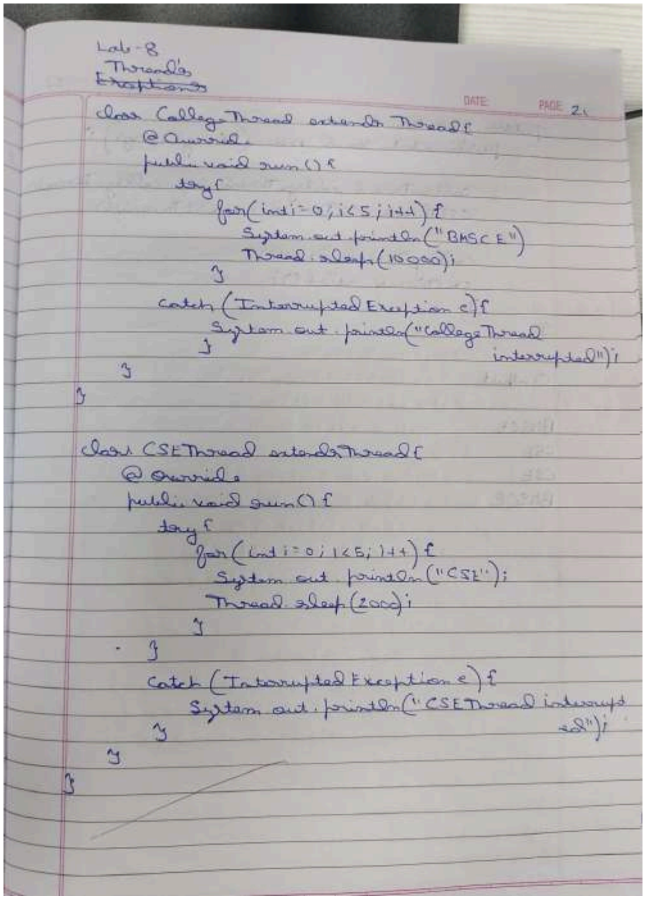


```
Enter Father's Age: 40
Enter Son's Age: 10
Father's Age: 40
Son's Age: 10
USN: 1BM23CS147
Name: Kedar Jevargi
kedarjevargi@Kedars-MacBook-Pro Code %  /usr/bin/env /Library/Java/JavaVirtual
pport/Code/User/workspaceStorage/b66d5e2f3a828121464e55353b6b2e73/redhat.java/
Enter Father's Age: -10
Exception Caught: Age cannot be negative
USN: 1BM23CS147
Name: Kedar Jevargi
kedarjevargi@Kedars-MacBook-Pro Code %  /usr/bin/env /Library/Java/JavaVirtual
pport/Code/User/workspaceStorage/b66d5e2f3a828121464e55353b6b2e73/redhat.java/
Enter Father's Age: 30
Enter Son's Age: -10
Exception Caught: Age cannot be negative
USN: 1BM23CS147
Name: Kedar Jevargi
kedarjevargi@Kedars-MacBook-Pro Code %  /usr/bin/env /Library/Java/JavaVirtual
pport/Code/User/workspaceStorage/b66d5e2f3a828121464e55353b6b2e73/redhat.java/
Enter Father's Age: 30
Enter Son's Age: 60
Exception Caught: Son's age cannot be greater than or equal to Father's age
USN: 1BM23CS147
Name: Kedar Jevargi
kedarjevargi@Kedars-MacBook-Pro Code % 
```

Program 8

Threads

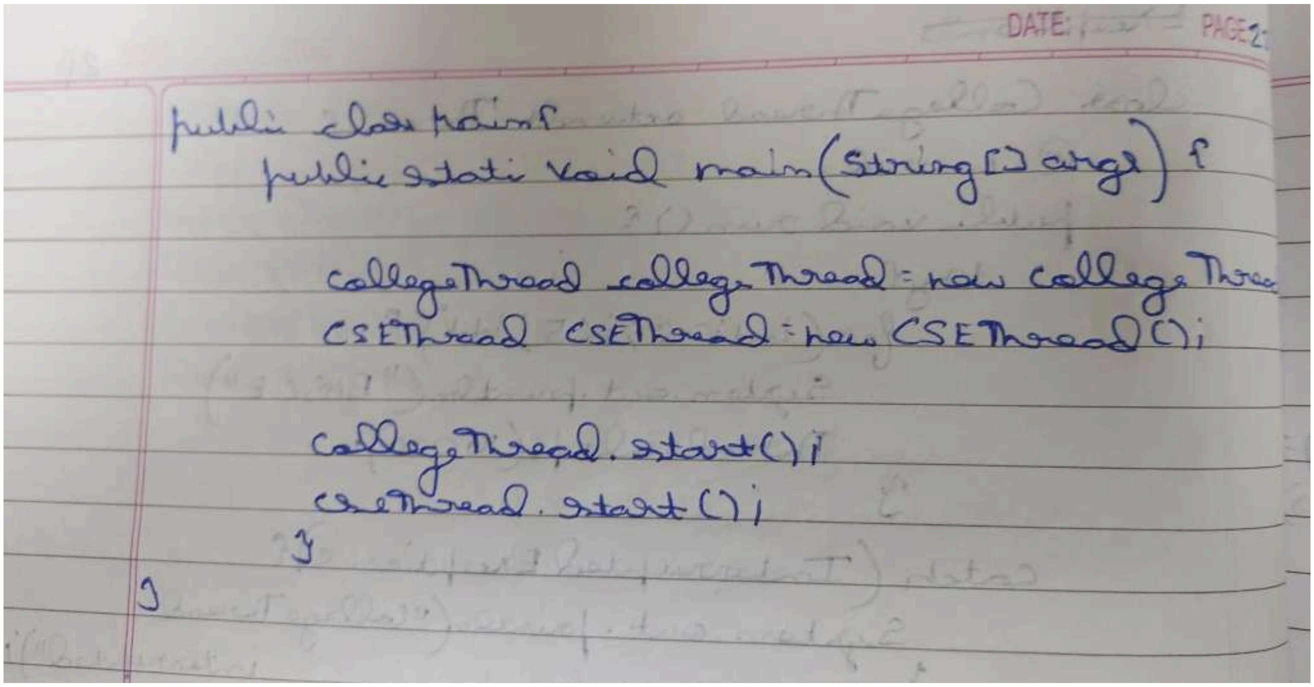
Algorithm:



Lab-8
Thread's
Exceptions

DATE: PAGE 21

```
class CollegeThread extends Thread{
    @Override
    public void run(){
        try{
            for(int i=0;i<5;i++){
                System.out.println("BMSCE")
                Thread.sleep(10000);
            }
        catch(InterruptedException e){
            System.out.println("CollegeThread
                                    interrupted");
        }
    }
}

class CSEThread extends Thread{
    @Override
    public void run(){
        try{
            for(int i=0;i<5;i++){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException e){
            System.out.println("CSEThread interrupt
                                    ed");
        }
    }
}
```

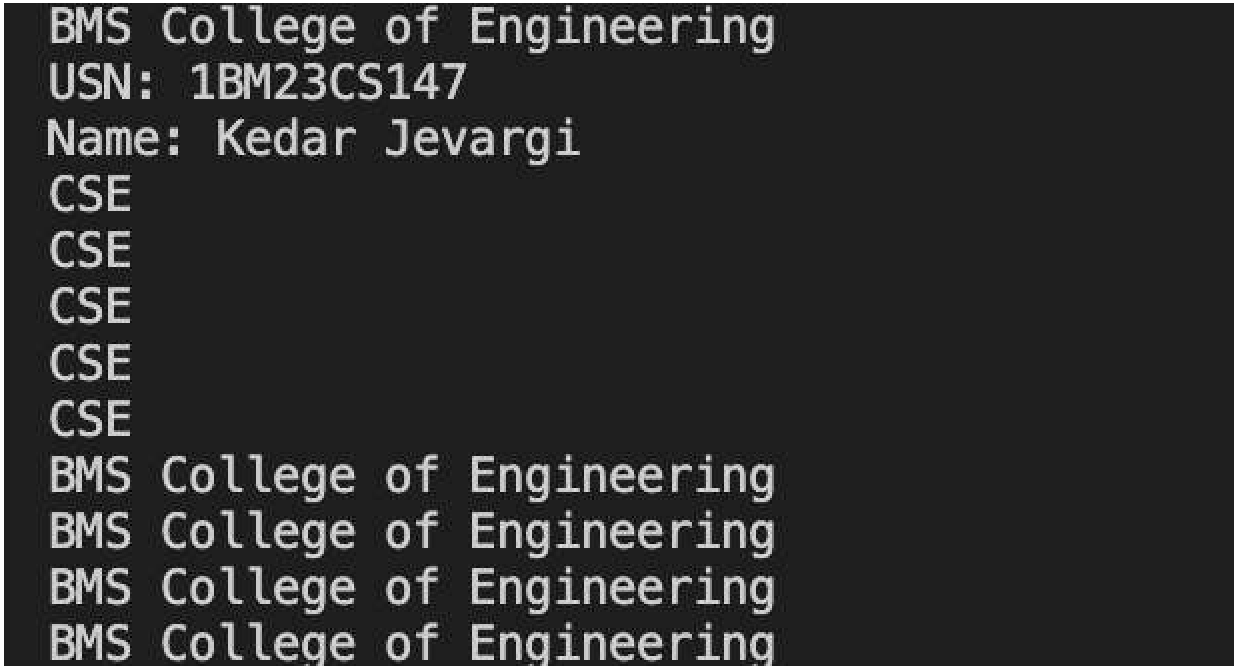


DATE: PAGE:

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

        CollegeThread collegeThread = new CollegeThrea
        CSEThread CSEThread = new CSEThread();

        collegeThread.start();
        cseThread.start();
    }
}
```

Code:

```
class CollegeThread extends Thread {
   @Override
   public void run() {
      try {
         for (int i = 0; i < 5; i++) {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         }
      } catch (InterruptedException e) {
         System.out.println("CollegeThread interrupted");
      }
   }
}

class CSEThread extends Thread {
   @Override
   public void run() {
      try {
         for (int i = 0; i < 5; i++) {
            System.out.println("CSE");
            Thread.sleep(2000);
         }
      } catch (InterruptedException e) {
         System.out.println("CSEThread interrupted");
      }
   }
}

public class Main {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

        collegeThread.start();
        cseThread.start();

        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
    }
}
```

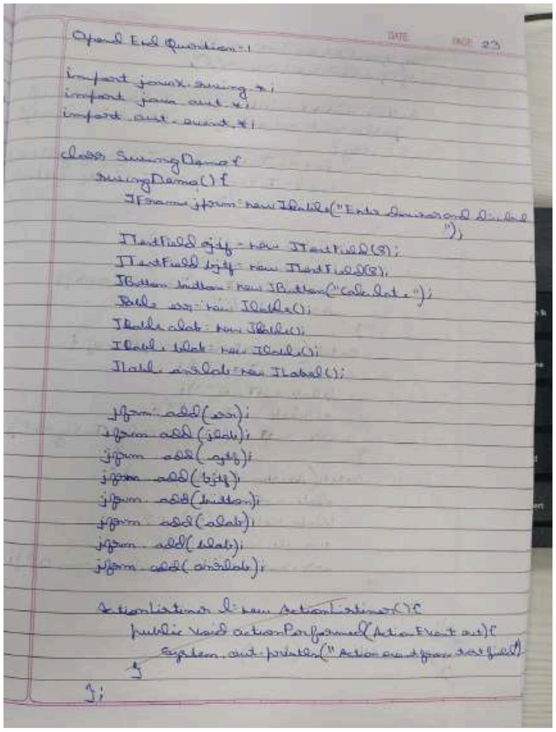


```
BMS College of Engineering
USN: 1BM23CS147
Name: Kedar Jevargi
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

Program 9

Open End Question 1

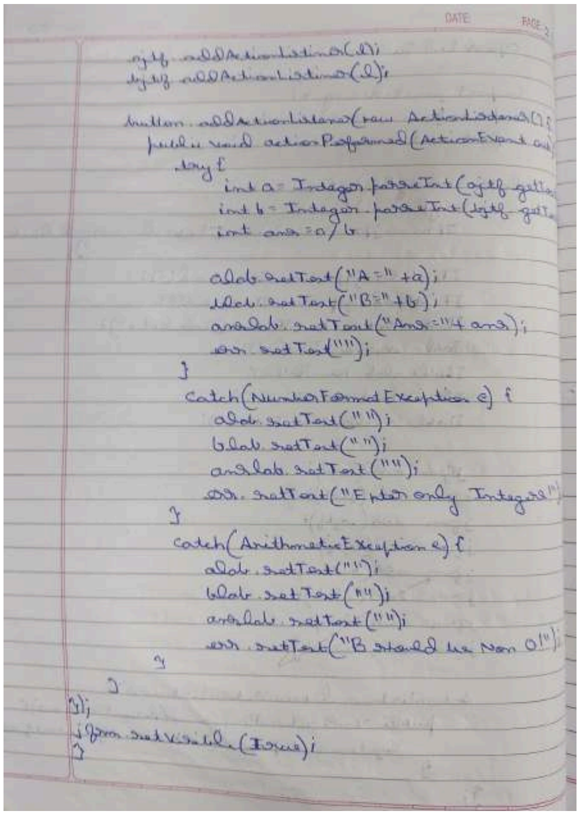
Algorithm:



DATE PAGE 23

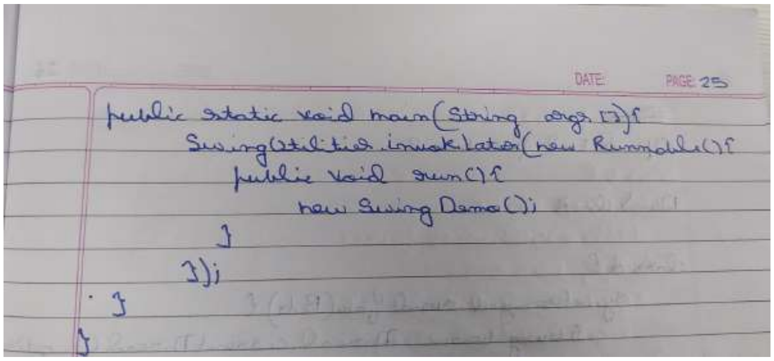
Open End Question-1

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

class SwingDemo{
    SwingDemo(){
        JFrame jform = new JLabel("Enter Divisor and Dividend
                                                    ");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate.");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jform.add(err);
       jform.add(jlab);
       jform.add(ajtf);
       jform.add(bjtf);
       jform.add(button);
       jform.add(alab);
       jform.add(blab);
       jform.add(anslab);

       ActionListener l = new ActionListener(){
           public void actionPerformed(ActionEvent evt){
               System.out.println("Action event from text field")
           }
   };
```



DATE PAGE 2

```
ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener(){
  public void actionPerformed(ActionEvent ae
    try{
        int a= Integer.parseInt(ajtf.getTe
        int b= Integer.parseInt(bjtf.getT
        int ans=a/b;

       alab.setText("A="+a);
       blab.setText("B="+b);
       anslab.setText("Ans="+ ans);
       err.setText("");
    }
     catch(NumberFormatException e) {
       alab.setText("");
       blab.setText("");
       anslab.setText("");
       err.setText("Enter only Integers")
    }
    catch(ArithmeticException e) {
       alab.setText("");
       blab.setText("");
       anslab.setText("");
       err.setText("B should be Non 0!");
    }
  }
});
jfrm.setVisible(true);
}
```

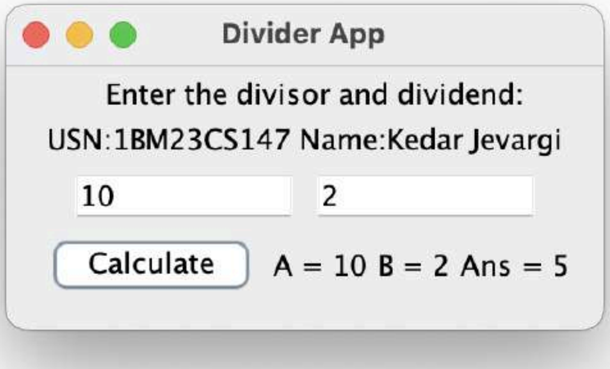


DATE: PAGE: 25

```
public static void main(String args[]){
      SwingUtilities.invokeLater(new Runnable(){
          public void run(){
               new SwingDemo();
          }
      });
 }
}
```

Code:

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

class SwingDemo {
   SwingDemo() {
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JLabel jlab = new JLabel("Enter the divisor and dividend:");
      JLabel jlab1 = new JLabel("USN:1BM23CS147 Name:Kedar Jevargi");
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);
      JButton button = new JButton("Calculate");
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      jfrm.add(err);
      jfrm.add(jlab);
      jfrm.add(jlab1);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      ActionListener l = new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            System.out.println("Action event from a text field");
         }
      };

      ajtf.addActionListener(l);
      bjtf.addActionListener(l);

      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            try {
               int a = Integer.parseInt(ajtf.getText());
               int b = Integer.parseInt(bjtf.getText());
               int ans = a / b;
               alab.setText("A = " + a);
               blab.setText("B = " + b);
               anslab.setText("Ans = " + ans);
               err.setText("");
            } catch (NumberFormatException e) {
               alab.setText("");
               blab.setText("");
               anslab.setText("");
               err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
               alab.setText("");
               blab.setText("");
               anslab.setText("");
               err.setText("B should be NON zero!");
            }
         }
      });

      jfrm.setVisible(true);
   }

   public static void main(String args[]) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            new SwingDemo();
         }
      });
   }
}
```



Enter the divisor and dividend:

USN:1BM23CS147 Name:Kedar Jevargi

10

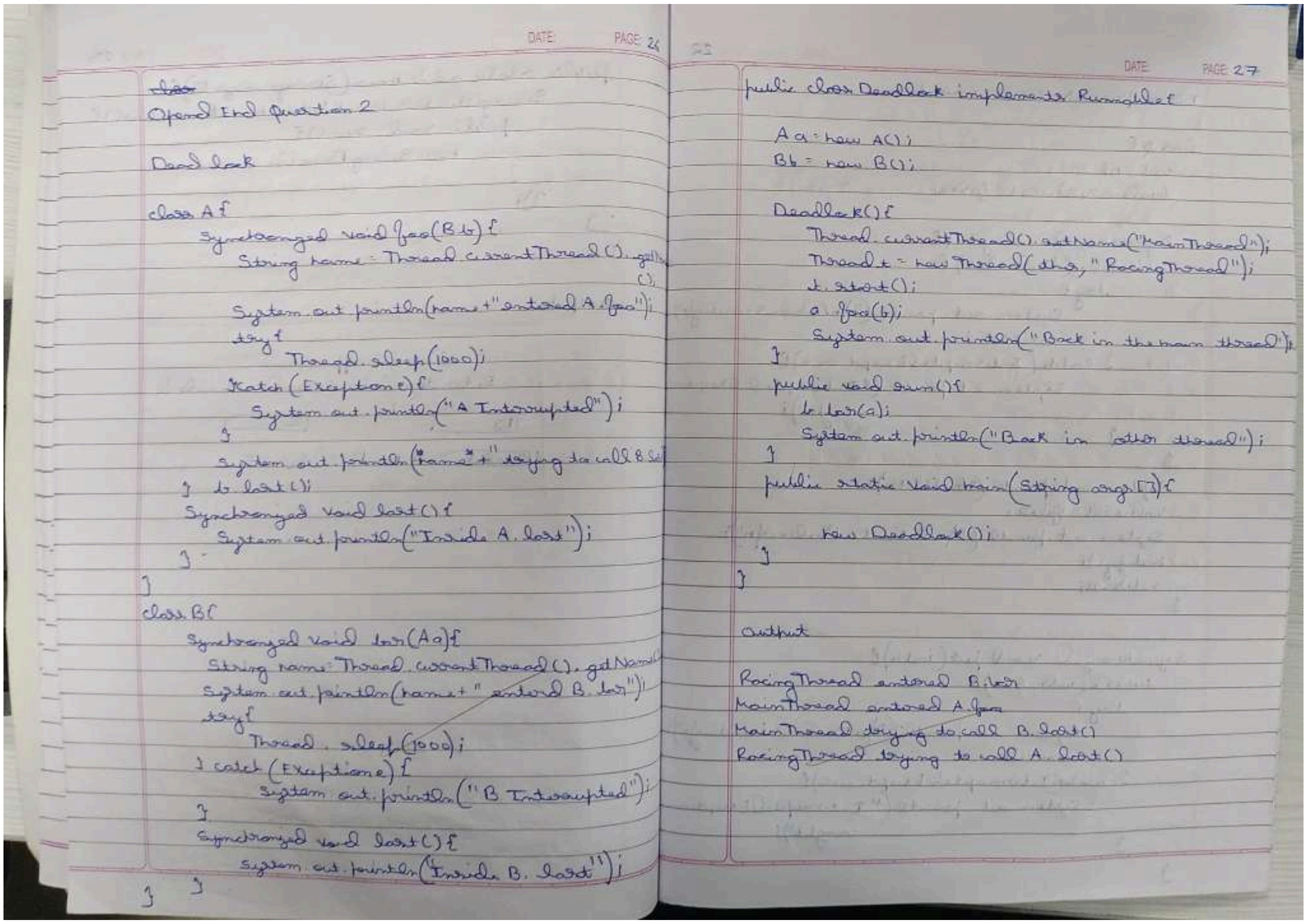
2

A = 10 B = 2 Ans = 5

Program 10

Open End Question 2

Algorithm:



DATE PAGE 26

~~class~~

Opend End Question 2

Dead lock

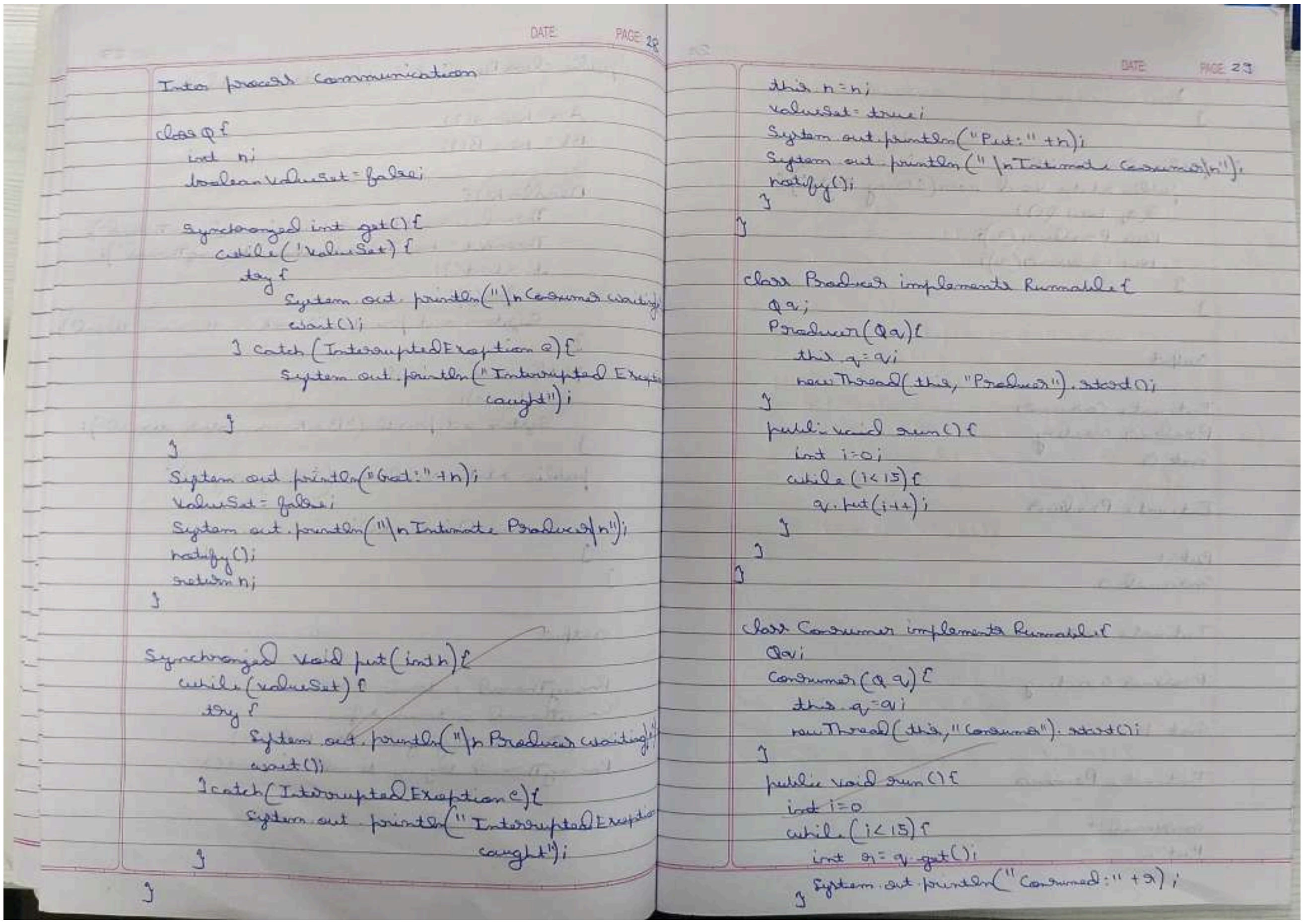
```
class A{
    Synchronized void foo(B b){
        String name = Thread.currentThread().getN
                                               ();
        System.out.println(name+" entered A.foo");
        try{
            Thread.sleep(1000);
        }catch(Exception e){
            System.out.println("A Interrupted");
        }
        System.out.println(name+" trying to call B.last
    } b.last();
    Synchronized void last(){
        System.out.println("Inside A.last");
    }
}
class B{
    Synchronized void bar(A a){
        String name=Thread.currentThread().getName(
        System.out.println(name+" entered B.bar");
        try{
            Thread.sleep(1000);
        } catch(Exception e){
            System.out.println("B Interrupted");
        }
        Synchronized void last(){
            System.out.println("Inside B.last");
}   }
```

DATE PAGE 27

```
public class Deadlock implements Runnable{

    A a = new A();
    B b = new B();

    Deadlock(){
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in the main thread");
    }
    public void run(){
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]){

        new Deadlock();
    }
}
```

Output

```
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
```



DATE: PAGE 22

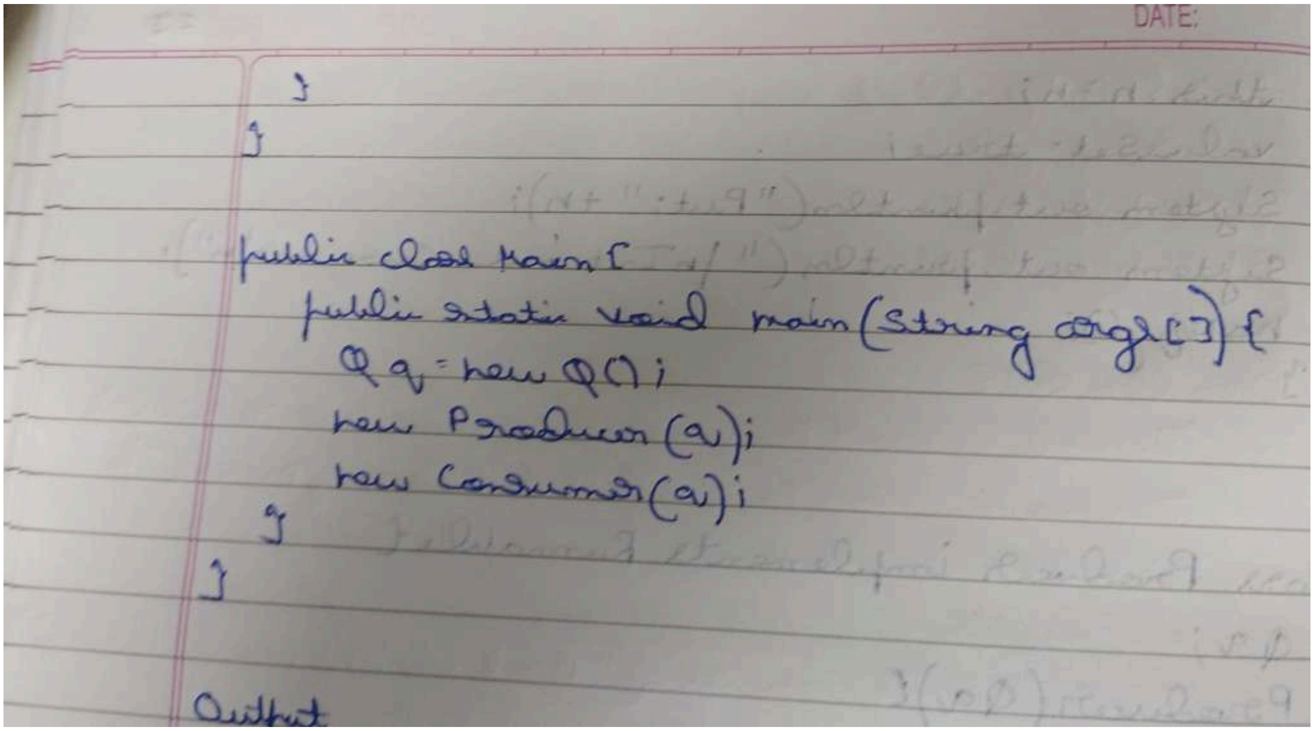
Inter process communication

```
class Q{
  int n;
  boolean valueSet=false;

  Synchronized int get(){
    while(!valueSet){
      try{
        System.out.println("\nConsumer waiting
        wait();
      } catch (InterruptedException e){
        System.out.println("Interrupted Excepti
                              caught");
      }
    }
    System.out.println("Got:"+n);
    valueSet=false;
    System.out.println("\nIntimate Producer\n");
    notify();
    return n;
  }

  Synchronized void put(int n){
    while(valueSet){
      try{
        System.out.println("\nProducer waiting")
        wait();
      }catch(InterruptedException e){
        System.out.println("InterruptedExceptio
                              caught");
      }
```

```
}
```

DATE: PAGE 23

```
    this.n=n;
    valueSet=true;
    System.out.println("Put:"+n);
    System.out.println("\nIntimate Consumer\n");
    notify();
  }
}

class Producer implements Runnable{
  Q q;
  Producer(Q q){
    this.q=q;
    new Thread(this,"Producer").start();
  }
  public void run(){
    int i=0;
    while(i<15){
      q.put(i++);
    }
  }
}

class Consumer implements Runnable{
  Q q;
  Consumer(Q q){
    this.q=q;
    new Thread(this,"Consumer").start();
  }
  public void run(){
    int i=0
    while(i<15){
      int r=q.get();
      System.out.println("Consumed:"+r);
    }
```



DATE:

```
    }
  }

public class Main{
  public static void main(String args[]){
    Q q=new Q();
    new Producer(q);
    new Consumer(q);
  }
}
```

Output

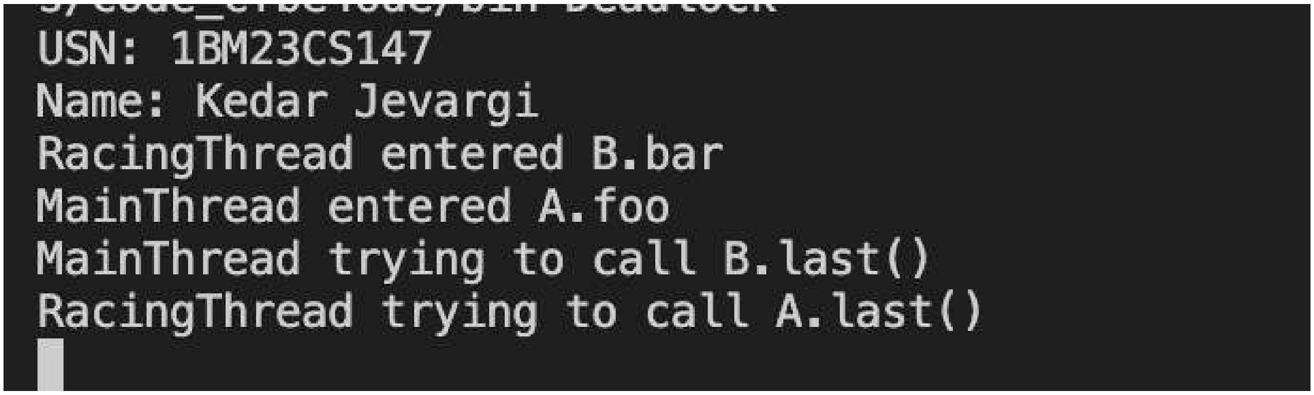
Code:

```
class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    synchronized void last() {
        System.out.println("Inside A.last");
    }
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    synchronized void last() {
        System.out.println("Inside B.last");
    }
}

public class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        System.out.println("USN: 1BM23CS147");
        System.out.println("Name: Kedar Jevargi");
        new Deadlock();
    }
}
```

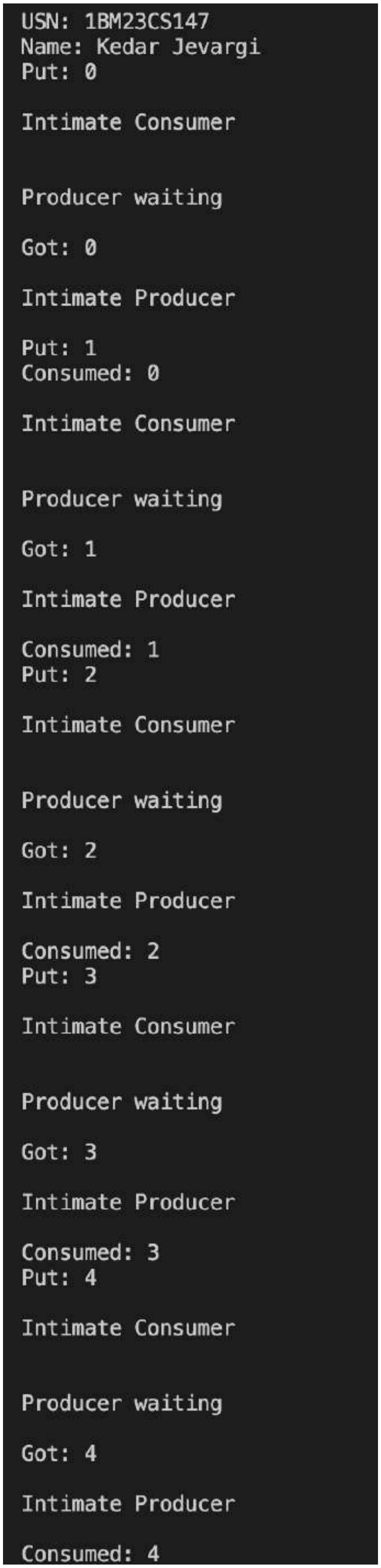
```
class Q {
    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (!valueSet) {
            try {
                System.out.println("\nConsumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nIntimate Producer\n");
        notify();
        return n;
    }

    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("\nProducer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
```

```
         }
      }
      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      System.out.println("\nIntimate Consumer\n");
      notify();
   }
}

class Producer implements Runnable {
   Q q;

   Producer(Q q) {
      this.q = q;
      new Thread(this, "Producer").start();
   }

   public void run() {
      int i = 0;
      while (i < 15) {
         q.put(i++);
      }
   }
}

class Consumer implements Runnable {
   Q q;

   Consumer(Q q) {
      this.q = q;
      new Thread(this, "Consumer").start();
   }

   public void run() {
      int i = 0;
      while (i < 15) {
         int r = q.get();
         System.out.println("Consumed: " + r);
         i++;
      }
   }
}

public class Main {
   public static void main(String args[]) {
      Q q = new Q();
```

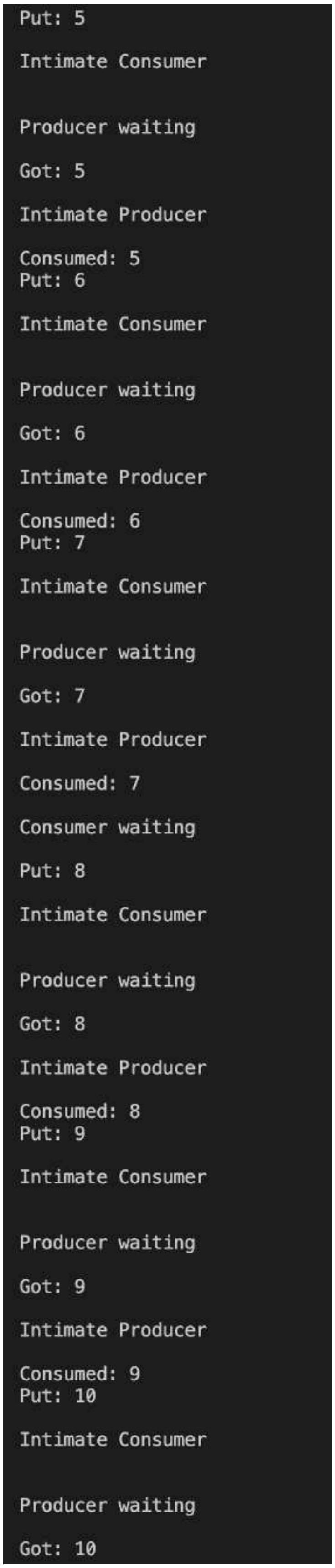
```
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("\nUSN: 1BM23CS147\nName: Kedar Jevargi");
    }
}
```



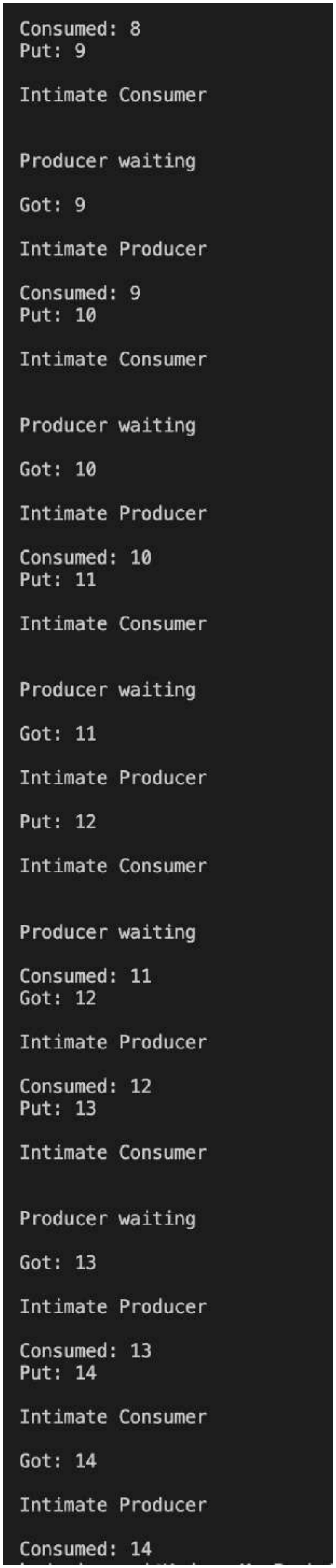
```
USN: 1BM23CS147
Name: Kedar Jevargi
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
```



```
USN: 1BM23CS147
Name: Kedar Jevargi
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1
Consumed: 0

Intimate Consumer


Producer waiting

Got: 1

Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

Consumed: 2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

Consumed: 3
Put: 4

Intimate Consumer


Producer waiting

Got: 4

Intimate Producer

Consumed: 4
```



```
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

Consumed: 5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

Consumed: 6
Put: 7

Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

Consumed: 7

Consumer waiting

Put: 8

Intimate Consumer


Producer waiting

Got: 8

Intimate Producer

Consumed: 8
Put: 9

Intimate Consumer


Producer waiting

Got: 9

Intimate Producer

Consumed: 9
Put: 10

Intimate Consumer


Producer waiting

Got: 10
```



```
Consumed: 8
Put: 9

Intimate Consumer


Producer waiting

Got: 9

Intimate Producer

Consumed: 9
Put: 10

Intimate Consumer


Producer waiting

Got: 10

Intimate Producer

Consumed: 10
Put: 11

Intimate Consumer


Producer waiting

Got: 11

Intimate Producer

Put: 12

Intimate Consumer


Producer waiting

Consumed: 11
Got: 12

Intimate Producer

Consumed: 12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

Consumed: 13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

Consumed: 14
```