WEEK-3

1) Develop a Java program that print all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c & use the quadratic formula. If the discriminate b^2-4ac is '-'ve, display a message stating that there are no real solution.

```
import java.lang.Math;
import java.util.*;

class quadraticEquation {
    public static void main (String[] args){
    Scanner s = new Scanner (System.in);
    Double a, b, c, det, r1, r2;
    System.out.println ("Enter 3 numbers in
     a quadratic equation.");
    a = s.nextDouble();
    b = s.nextDouble();
    c = s.nextDouble();
    det = (b*b) - (4*a*c);
     if (det > 0) {
       r1 = (-b + Math.Sqrt(det)) / (2*a);
       r2 = (-b - Math.Sqrt (det)) / (2*a);
    System.out.println(" Roots are real and
    unequal. \n Root : " +r1+ ", " +r2);
    }
     else if (det == 0) {
       r1 = -b / (2*a);
     System.out.println ("Roots are real and
       equal. \n Roots: " +r);
     }
     else {
        System.out.println ("No real root.");
     }
}
```

2 Accept an array of n integers. Find the number of positive numbers, negative numbers and zeros.

```
import java.util.*;

class noOfPNZ {
    public static void main(String [] args) {
        Scanner s = new Scanner (System.in);
        int [] a = new int[10];
        int n, pCount = 0, nCount = 0, ZCount = 0.
        System.out.println ("Enter size of array.");
        n = s.nextInt();
        System.out.println ("Enter elements of array.");
        for (int i = 0 ; i < n ; i++) {
            a [i] = s.nextInt();
            If (a [i] > 0)
                pCount ++;
            else if (a[i] < 0)
                nCount ++ ;
            else
                ZCount ++;
        }

        System.out.println ("Number of positive
                values : " + pCount);
        System.out.println("Number of negative
                values : " + nCount);
        System.out.println ("Number of zeros : "
                + ZCount);

    }
}
}
```

3 Accept an array of size n from the user. Find the sum of even indices (i.e. 0, 2, 4, ---) & sum of odd indices (1, 3, 5) & print the same.

```
import java.util.*;

Class SumOfIndices {
    public static void main (String[] args) {
        Scanner s = new Scanner (System.in);
        int[] a = new int[10];
        int n, oddIndexSum=0, evenIndexSum=0;
        System.out.println ("Enter size of array.");
        n = s.nextInt();
        System.out.println ("Enter element of array.");

        for (int i=0; i<n; i++){
            a[i] = s.nextInt();

            if (i%2 == =0)
                evenIndexSum = evenIndexSum + a[i];
            else
                oddIndexSum = oddIndexSum + a[i];
        }

        System.out.println ("Sum of even index
            values : " + evenIndexSum);
        System.out.println ("Sum of odd index
            values : " + oddIndexSum);
    }
}
```

Q Consider a super market bill. Accept a double array holding rate per item of say x items & and an int array showing the quantity purchased by a customer. Calculate the total bill amount & the final bill amount after giving discounts as per the following slabs.

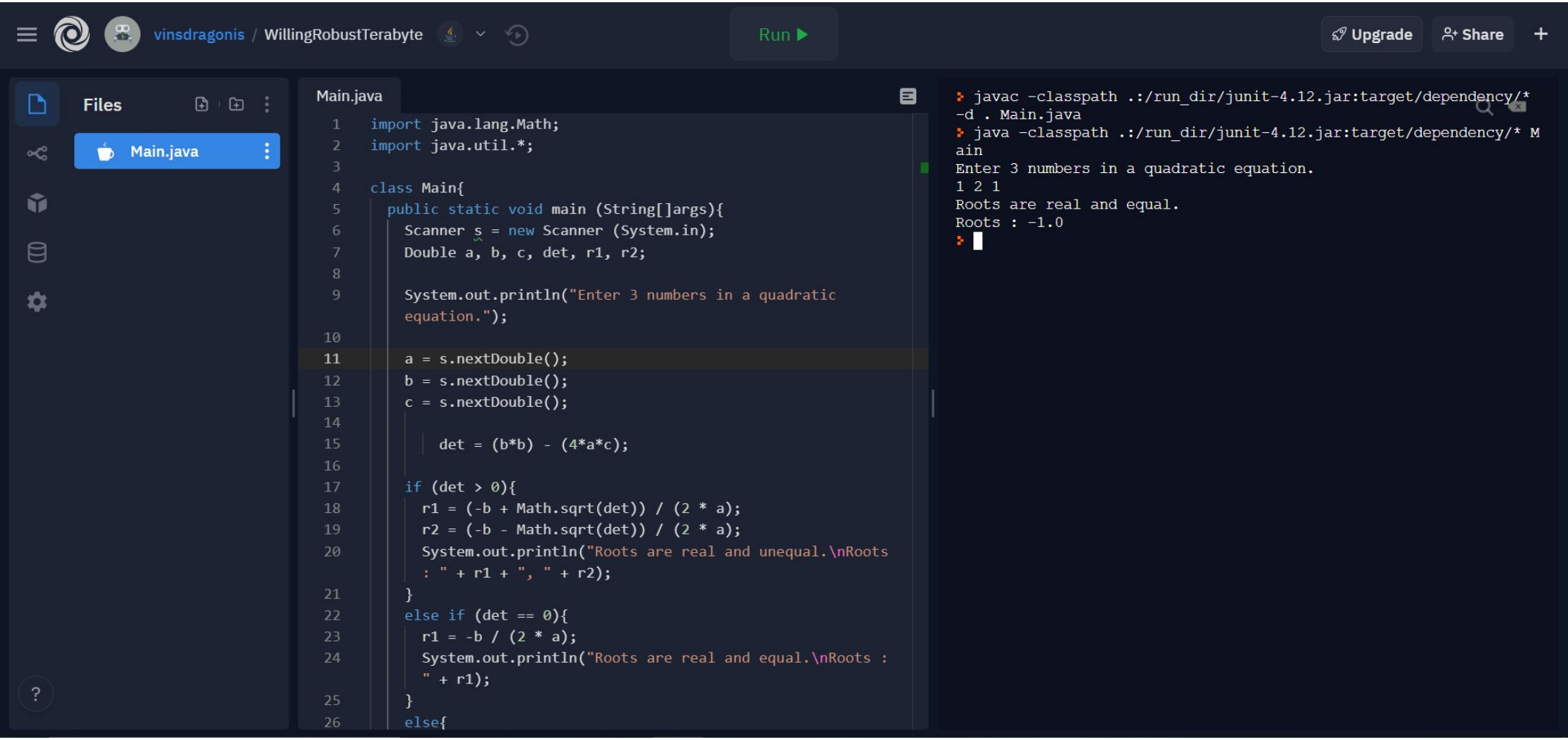
If the total bill amount >= 10000, discount = 5%.
If the total bill amount >= 7500 & < 10000, discount = 3%.
If the total bill amount >= 5000, discount = 2%.

```
import java.util.*;

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

     int [] [] a = new int [10] [2];
     double bill = 0, discount, items, totalBill;
     System.out.println ("Enter number of
        items.");
     item = s.nextInt ();
     System.out.println ("Enter quantity and
       price of item.");

     for (int i=0; i<items; i++) {
       for (int j=0; j<2; j++)
     a [i] [j] = s.nextInt ();
     }

     for (int i=0; i<items; i++) {
     int quantity = a [i] [0];
     int pricePerItem = a [i] [1];
```

```
        bill += quantity * pricePerItem;
    }

    System.out.println("Bill = " + bill);

    if (bill >= 10000)
        discount = 0.05D;
    else if (bill < 10000 && bill >= 7500)
        discount = 0.03D;
    else
        discount = 0.02D;

    totalBill = bill * discount;

    System.out.println("Bill after discount: "
                        + totalBill);
    }
}
```

5 Accept an array A of n elements. Create two new arrays where the first one say B that holds all the odd numbers from array A & the second say C holds the even numbers from array D. Display the Sum, average, max & min of array C.

```
import java.util.*;

class arrayOrganizer{

    public static void main(String[] args){

        Scanner S= new Scanner(System.in);

        int[] a = new int[10], b= new int [10],
              c= new int[10];
        int cSum=0, cAvg, cMin, cMax, j=0, k=0, n;

        System.out.println("Enter Size of array.");
        n= S.nextInt();
        System.out.println("Enter elements of array.");

        for(int i=0; i<n ; i++){
            a[i]= S.nextInt();

            If(a[i]% 2 !0){
        b[j]= a[i];
        j++;
        }
                -else{
            c[k]= a[i];
            k++;
        }
```


vinsdragonis / WillingRobustTerabyte
Run
Upgrade
Share
Files
Main.java
Main.java
1 import java.lang.Math;
2 import java.util.*;
3
4 class Main{
5 public static void main (String[]args){
6 Scanner s = new Scanner (System.in);
7 Double a, b, c, det, r1, r2;
8
9 System.out.println("Enter 3 numbers in a quadratic equation.");
10
11 a = s.nextDouble();
12 b = s.nextDouble();
13 c = s.nextDouble();
14
15 det = (b*b) - (4*a*c);
16
17 if (det > 0){
18 r1 = (-b + Math.sqrt(det)) / (2 * a);
19 r2 = (-b - Math.sqrt(det)) / (2 * a);
20 System.out.println("Roots are real and unequal.\nRoots : " + r1 + ", " + r2);
21 }
22 else if (det == 0){
23 r1 = -b / (2 * a);
24 System.out.println("Roots are real and equal.\nRoots : " + r1);
25 }
26 else{
javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d . Main.java
java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Main
Enter 3 numbers in a quadratic equation.
1 2 1
Roots are real and equal.
Roots : -1.0

week 2

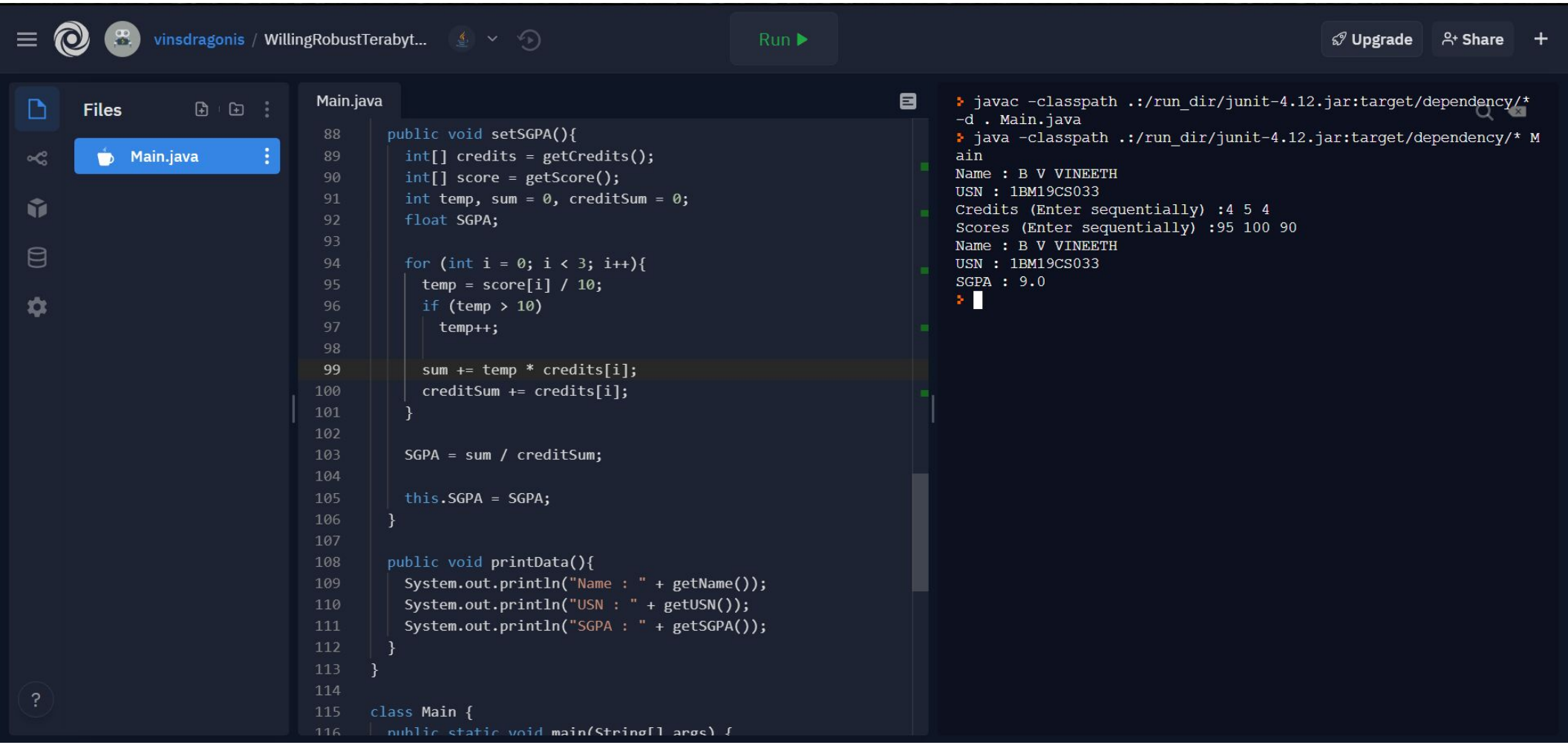
2) Develop a java program to create a class Student with members USN, name, an array credits & an an array marks. Include methods to accept & display details & a method to calculate SGPA of a Student.

```
import java.lang.*;
import java.util.Scanner;

Class Student {
    private String name;
    private String USN;
    private int[] credits;
    private int[] Score;
    private float SGPA;

    public Student() {
    int[] credits = new int[3];
    int[] Score = new int[3];
    }

    public String getName() {
    return name;
    }

    public void SetName() {
      String name;

      System.out.print("Name: ");
      Scanner s = new Scanner(System.in);

      name = s.nextLine();
      this.name = name;
    }
```

```
public String getUSN() {
    return USN;
}

public void setUSN() {
String USN;

System.out.print("USN: ");
Scanner s = new Scanner(System.in);

USN = s.nextLine();
this.USN = USN;
}

public int[] getCredits() {
    return credits;
}

public void SetCredits() {
 int[] credit = new int[3];

System.out.print("Credit (Enter Sequentially)");
Scanner s = new Scanner(System.in);

for (int i=0; i<3; i++)
    Credits[i] = s.nextInt();

 this.credit = credits;
}

public int[] getScore() {
    return score;
}
```

```
public void setScore() {
int[] score = new int[3];

System.out.print("Scores (Enter sequentially):");
Scanner s= new Scanner(System.in);

for (int i=0; i<3; i++)
  score[i] = s.nextInt();

  this.score = score;
}

 public float getSGPA(){
     return SGPA;
 }

public void setData() {
 setName();
 setUSN;
 setCredits();
 setScore();
 setSGPA();
}

public void setSGPA() {
   int[] credit = getCredits();
   int[] score = getScore();
   int temp, sum=0, creditSum=0;
   float SGPA;

   for (int i=0; i<3; i++) {
    temp= score[i] /10;
    if (temp >10)
     temp ++;
```

```
        sum += temp * credits[i];
        System.out.println(sum);
        creditSum += credits[i];
    }

    SGPA = sum / creditSum;

    this.SGPA = SGPA;
    }

    public void printData() {
      System.out.println("Name = " + getName());
      System.out.println("USN: " + getUSN());
      System.out.println("SGPA: " + getSGPA());
    }
}

class SGPA_Calculator {
    public static void main(String[] args){
    Student st = new Student();

    st.setData();
    st.printData();

    }
}
```


vinsdragonis / WillingRobustTerabyt...
Run
Upgrade
Share
Files
Main.java
Main.java
public void setSGPA(){
int[] credits = getCredits();
int[] score = getScore();
int temp, sum = 0, creditSum = 0;
float SGPA;
for (int i = 0; i < 3; i++){
temp = score[i] / 10;
if (temp > 10)
temp++;
sum += temp * credits[i];
creditSum += credits[i];
}
SGPA = sum / creditSum;
this.SGPA = SGPA;
}
public void printData(){
System.out.println("Name : " + getName());
System.out.println("USN : " + getUSN());
System.out.println("SGPA : " + getSGPA());
}
}
class Main {
javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d . Main.java
java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Main
Name : B V VINEETH
USN : 1BM19CS033
Credits (Enter sequentially) :4 5 4
Scores (Enter sequentially) :95 100 90
Name : B V VINEETH
USN : 1BM19CS033
SGPA : 9.0

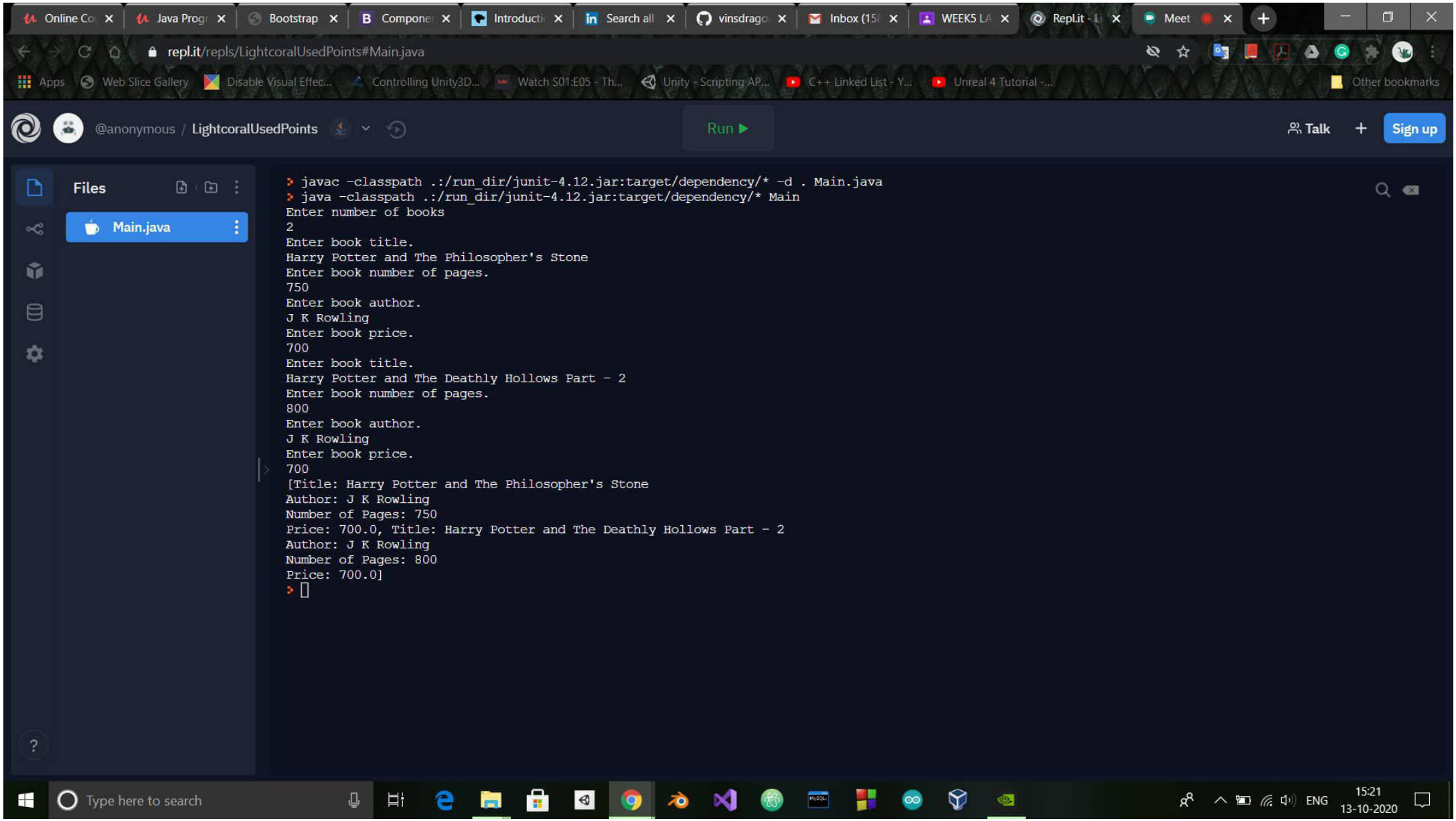
13/10/2022

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 & get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.*;
import java.util.ArrayList;
import java.util.List;

class book {
    private String title;
    private int noOfpages;
    private String author;
    private float price;

    public book() {
     this("unknown book", "Unknown author");
    }

    public book(String title, String author){
     this(title, 50, author, 0.0f);
    }

    public book(String title, int noOfpages,
                String author, float price) {
     setTitle(title);
     setNoOfpages(noOfpages);
     setAuthor(author);
     setprice(price);
    }
```

```
public String getTitle() {
    return title;
}

public void setTitle(String title){
    Scanner s = new Scanner(System.in);

    System.out.println("Enter book title.");
    title = s.nextLine();
    this.title = title;
}

public int getNoOfpages() {
    return noOfpages;
}

public Void SetNoOfpages(int noOfpages) {
    Scanner s= new Scanner(System.in);

    System.out.println("Enter book number of
                        pages.");
    noOfpages = s.nextInt();
    this.noOfpages = noOfpages;
}

public String getAuthor() {
    return author;
}

public Void SetAuthor(String author) {
    Scanner s= new Scanner(System.in);

    System.out.println("Enter book author.");
    author = s.nextLine();
```

```
        this.author = author;
    }

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
        Scanner s = new Scanner(System.in);

        System.out.println("Enter book price.");
        price = s.nextInt();
        this.price = price;
    }

    @Override
    public String toString() {
        return "Title: " + getTitle() + "\nAuthor: " +
        getAuthor() + "\nNumber of pages: " +
        getNoOfpages() + "\nPrice: " + getPrice() + "\n";
    }
}

class bookDetails {
    public static void main(String[] args) {
        ArrayList<book> b = new ArrayList<book>();
        Scanner s = new Scanner(System.in);
        int n, i;
        System.out.println("Enter number of books");
        n = s.nextInt();
        i = 0;
        while (i<n) {
        b.add(new book());
        i++;
        }
        System.out.println(b.toString());
    }
```


repl.it/repls/LightcoralUsedPoints#Main.java
@anonymous / LightcoralUsedPoints
Run
Talk
Sign up
Files
Main.java
javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d . Main.java
java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Main
Enter number of books
2
Enter book title.
Harry Potter and The Philosopher's Stone
Enter book number of pages.
750
Enter book author.
J K Rowling
Enter book price.
700
Enter book title.
Harry Potter and The Deathly Hollows Part - 2
Enter book number of pages.
800
Enter book author.
J K Rowling
Enter book price.
700
[Title: Harry Potter and The Philosopher's Stone
Author: J K Rowling
Number of Pages: 750
Price: 700.0, Title: Harry Potter and The Deathly Hollows Part - 2
Author: J K Rowling
Number of Pages: 800
Price: 700.0]
Type here to search
ENG
15:21
13-10-2020

Week 6

Develop a Java program to find the transpose of a given matrix of order MxN

```
import java.util.*;

class transposeOfMatrix {
  public static void main(String[] args) {
   int m, n, i = 0, j=0
   Scanner s= new Scanner(System.in);

   System.out.println("Enter number of columns
     in matrix");
   n= s.nextInt();

   int[][] a = new int[m][n];
   int [][] b = new int [n][m];

  System.out.println("Enter element of matrix:");

   for (i=0 ; i <m ; i++) {
     for (j=0; j<n ; j++)
      a [i][j] =s.nextInt();
    }

    for (i=0 ; i <m; i++) {
      for(j =0 ; j<n ; j++)
       b[j][i] = a[i][j];
    }

    System.out.println("n original matrix: ");

    for (i=0 ; i<m ; i++) {
      for (j=0 ; j<n ; j++)
        System.out.print(a[i][j] + " ");
        System.out.println();
```

```
System.out.println("\n Transpose of original
        matrix: ");

for (i=0; i<n; i++) {
    for (j=0; j<m; j++)
        System.out.print(b[i][j] + " ");
        System.out.println();
    }
}
```

2. Develop a Java program which has the (only) circleDemo that has members - radius, area & perimeter. Include methods to do the following
   a. accept the radius from the User
   b. find the area of the circle
   c. find the perimeter of the circle
   d. Display all the details.

```
import java.util.*;
import java.lang.Math.*;

class CircleDemo {
  plubic static void main(String[] args) {
    double r, P, Ar;
    Scanner S=new Scanner(System.in);

    System.out.println("Enter radius");
    r= S.nextDouble();

    P= 2*Math.PI*r;
    Ar= Math.PI * (Math.pow(r,2));

    System.out.println("perimeter = " + P + "\n
    Area = " + Ar);
  }
}
```

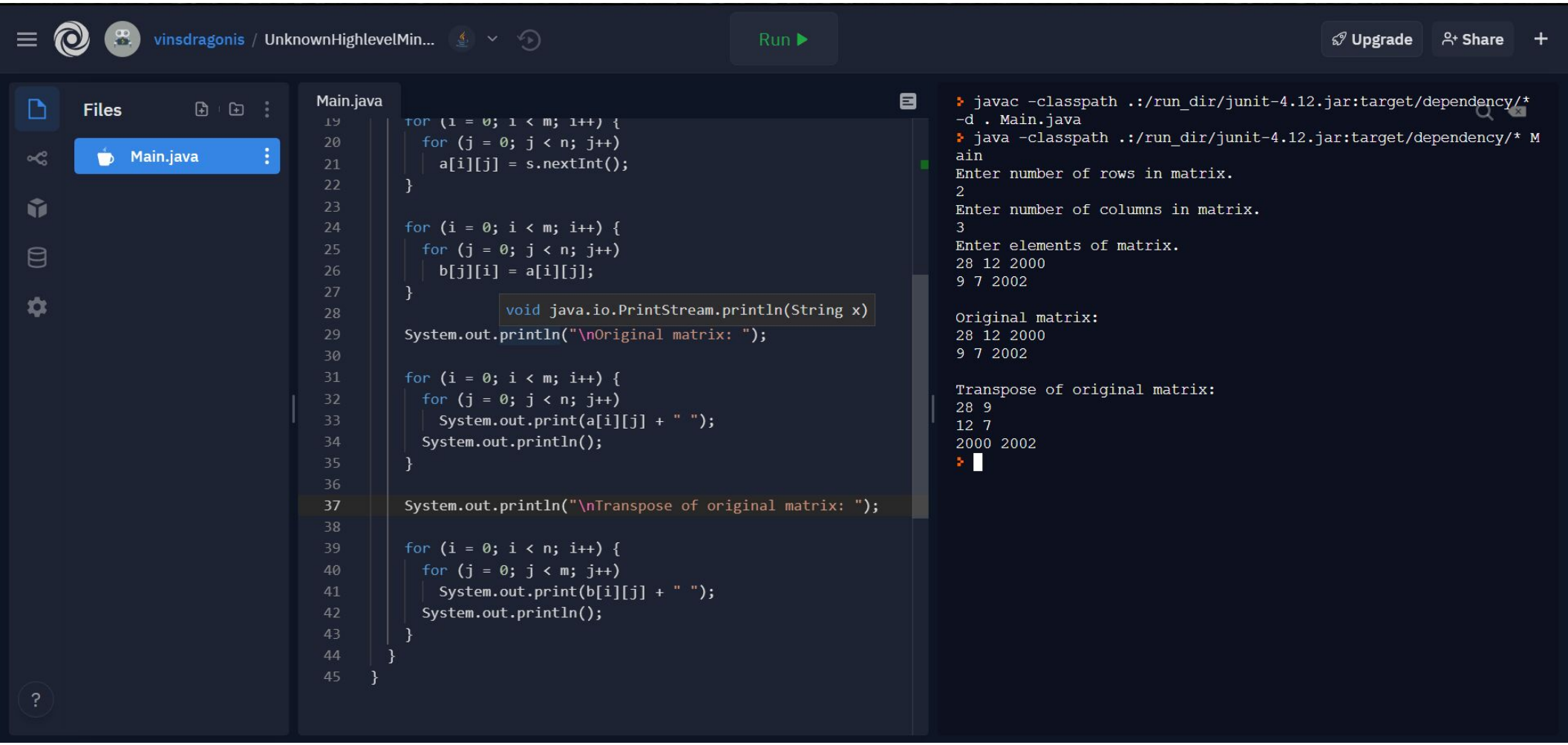
3 Develop a Java program to create a class Actor with id, name, no-of_movies, no-of_years-exp. Calculate the average performance for each of the actor & print the name of the actor with highest average.

```
import java.util.*;

class Actor {
    private int ID;
    private String name;
    private int no-of_movies;
    private int no-of_years_exp;
    private double avgperformance;
    Scanner S= new Scanner(System.in);

    public Actor() {
        setData();
    }

    public int getID() {
      return this.ID;
    }

    public void setID() {
      int id = S.nextInt();
      this.ID = id;
    }

    public String getName() {
      return this.name;
    }

    public void setName() {
```

```
String name = S.nextline();
this.name = name;
}

public int getNo of Movies() {
  return this.no_of_movies;
}

public void SetNo of Movies() {
  int noOfMovies = S.nextInt();
  this.no_of_movies = noOfMovies;
}

public int getYearsOf Exp() {
  return this.no_of_years_exp;
}

public void setYearsOfExp() {
  int yearsOf Exp = S.nextInt();
  this.no_of_years_exp = yearsof Exp;
}

public Void SetData() {
  System.out.println("Enter actor ID.");
  setID();
System.out.println("Enter actor name.");
  setName();
  System.out.println("Enter number of movies.");
setNoOf Movies();
System.out.println("Enter years of experience");
  setYearsOFExp();
}
```

```
public double getAvgperformonce() {
    return this.avgperformance;
}

public Void SetAvgPerformonce() {
 this.avgperformance = getNoOFMovies/getYearsOf
}

public void printData() {
 getAvgPerformonce();

System.out.println("Actor ID: " + getID() +
\n Name: " + getName() + "\n Number of
movies: " + getNoOFMovies() + "\n Years
of Experience: " + getYearsOFExp() + "
\n Average: " + getAvgperformance());
}
}

 claxx actor {
 public static void main(String[] args){
  int n, i, highestIndex = 0;
  double highestAvg = 0.
  Scanner S = new Scanner(System.in);

 System.out.println("Enter number of
                    actors.");
  n = S.nextInt();

 Actor[] act = new Actor[n];

 for (i=0; i<n; i++) {
  act[i] = new Actor();
  act[i].printData();
 }
```

```
for(i=0; i<n; i++){
if(act[i].getAvgperformance()>highestAvg){
  highestAvg=act[i].getAvgperformance();
  highestIndex=i;
}
}

act[highestIndex].printData();
}
}
```


vinsdragonis / UnknownHighlevelMin...
Run
Upgrade
Share
Files
Main.java
Main.java
for (i = 0; i < m; i++) {
for (j = 0; j < n; j++)
a[i][j] = s.nextInt();
}
for (i = 0; i < m; i++) {
for (j = 0; j < n; j++)
b[j][i] = a[i][j];
}
void java.io.PrintStream.println(String x)
System.out.println("\nOriginal matrix: ");
for (i = 0; i < m; i++) {
for (j = 0; j < n; j++)
System.out.print(a[i][j] + " ");
System.out.println();
}
System.out.println("\nTranspose of original matrix: ");
for (i = 0; i < n; i++) {
for (j = 0; j < m; j++)
System.out.print(b[i][j] + " ");
System.out.println();
}
}
}
javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d . Main.java
java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Main
Enter number of rows in matrix.
2
Enter number of columns in matrix.
3
Enter elements of matrix.
28 12 2000
9 7 2002
Original matrix:
28 12 2000
9 7 2002
Transpose of original matrix:
28 9
12 7
2000 2002

27/10/2020

classmate Date Page

Week 7 - INHERITANCE PRACTICE.

1 Develop a java program to create a cl Student whose variables are usn, name & se Derive a class test from Student. It include an array of cie marks of each course & their corresponding credits in another array. Derive a class Exam from Test which includes an array of see mark Derive a class Result which calculates the grade for each course & the SGPA Create n students objects & displays all the above details.

```
import java.util.Scanner;

class Student {
    String USN, name;
    int sem;
    Scanner sc = new Scanner(System.in);
    void setStuDetails() {
     System.out.println("Enter USN of Student");
     this.USN = sc.nextLine();
     System.out.println("Enter Name of Student");
     this.name = sc.nextLine();
     System.out.println("Enter Semester of
                         Student:");
     this.sem = sc.nextInt();
    }

    Void getStudDetails() {
     System.out.println("USN:" + this.USN);
     System.out.println("Name:" + this.name);
     System.out.println("Semester:" + this.sem);
    }
}
```

```
class Test extends Student{
    double cieMarks[] = new double[5];
    int credits[] = new int[5];
    int totalCredits = 0;
    void setCieDetails() {
        for(int i=0; i<cieMarks.length; i++) {
            System.out.println("Enter CIE marks(50)
                    in Course" + (i+1) + ": ");
            cieMarks[i] = sc.nextDouble();
            System.out.println("Enter credits of Course" +
                    (i+1) + ":");
            credits[i] = sc.nextInt();
            totalCredits += credits[i];
        }
    }
}

class Exam extends Test{
    double seeMarks[] = new double[5];
    double totalMarks[] = new double[5];
    int totCredits = super.totalCredits;
    void setSeeDetails() {
        for(int i=0; i<cieMarks.length; i++){
            System.out.println("Enter SEE marks(100)
                    in course" + (i+1) + ": ");
            seeMarks[i] = sc.nextDouble()/2;
        }
        calcTotalMarks();
    }
    void calcTotalMarks() {
        for(int i=0; i<5; i++) {
            totalMarks[i] = cieMarks[i] + seeMarks[i];
        }
    }
}
```

```
class Result extends Exam {
    char grades[] = new char[5];
    double sgpa = 0;
    int points[] = new int[5];
    void calcSGPA() {
        for (int i = 0; i < 5; i++) {
            if (totalMarks[i] > 100) {
                System.out.println("Error : Marks are ab
                return;
            } else if (totalMarks[i] >= 90) {
                points[i] = 10;
            } else if (totalMarks[i] >= 80) {
                points[i] = 9;
            } else if (totalMarks[i] >= 70) {
                points[i] = 8;
            } else if (totalMarks[i] >= 60) {
                points[i] = 7;
            } else if (totalMarks[i] >= 50) {
                points[i] = 5;
            } else if (totalMarks[i] >= 40) {
                points[i] = 4;
            } else {
                points[i] = 0;
            }

            sgpa += (points[i] * credits[i]);
        }
    }

    void calcGrade() {
        for (int i=0 ; i<5 ; i++) {
            if (totalMarks[i] > 100) {
                System.out.println("Error: Marks are above
                return;
```

```
} else if (totalMarks[i] >= 90) {
        grades[i] = 'S';
    } else if (totalMarks[i] >= 80) {
        grades[i] = 'A';
    } else if (totalMarks[i] >= 70) {
        grades[i] = 'B';
    } else if (totalMarks[i] >= 60) {
        grades[i] = 'C';
    } else if (totalMarks[i] >= 50) {
        grades[i] = 'D';
    } else if (totalMarks[i] >= 40) {
        grades[i] = 'E';
    } else {
        points[i] = 'F';
    }
  }
}

void getSGPA() {
  System.out.format("SGPA is %.2f",
                    (sgpa / total(Credits)));
}

void getGrades() {
  for(int i=0; i<5; i++) {
    System.out.println("Subject "+(i+1)+":"+
                       grades[i]);
  }
}

public class SGPACalculator {
  public static void main(String[] args) {
    int n=0;
```

```
Scanner sc= new Scanner(System.in);
System.out.println("Enter Number of
                    Students");
n= sc.nextInt();
Result results[] = new Result[n];
for(int i=0; i<n; i++){
    results[i]= new Result();
    results[i].setStuDetails();
    results[i].setCieDetails();
    results[i].setSeeDetails();
    results[i].calcSgpa();
    results[i].calcGrade();
    results[i].getSgpa();
    results[i].getGrades();

  }
 }
}
```

Week 7 - Inheritance

classmate
Date
Page

2) Develop a Java program to create a class Player with member variables name, matches-played & average? This class has an abstract method cal-average (String, int, int). Derive two classes BATSMAN & BOWLER from PLAYER. Class BATSMAN has a member variable runs-scored. class BOWLER has a member variable runs-given. Create m BATSMAN objects & n BOWLER object. calculat & display the average runs scored by each BATSMAN & average runs given by each BOWLER.

```
import java.util.*;

abstract class player {
    String name;
    int matchesplayed;
    double avg;
    Scanner S = new Scanner(System.in);

abstract public String getName();
abstract public void SetName();
abstract public int getMatches().
abstract public void setMatches();
abstract public double getAvgScore();
abstract public void setAvgScore();
abstract public void printData();
}

class Batsman extends player {
    private double runsScored;

public Batsman() {
```

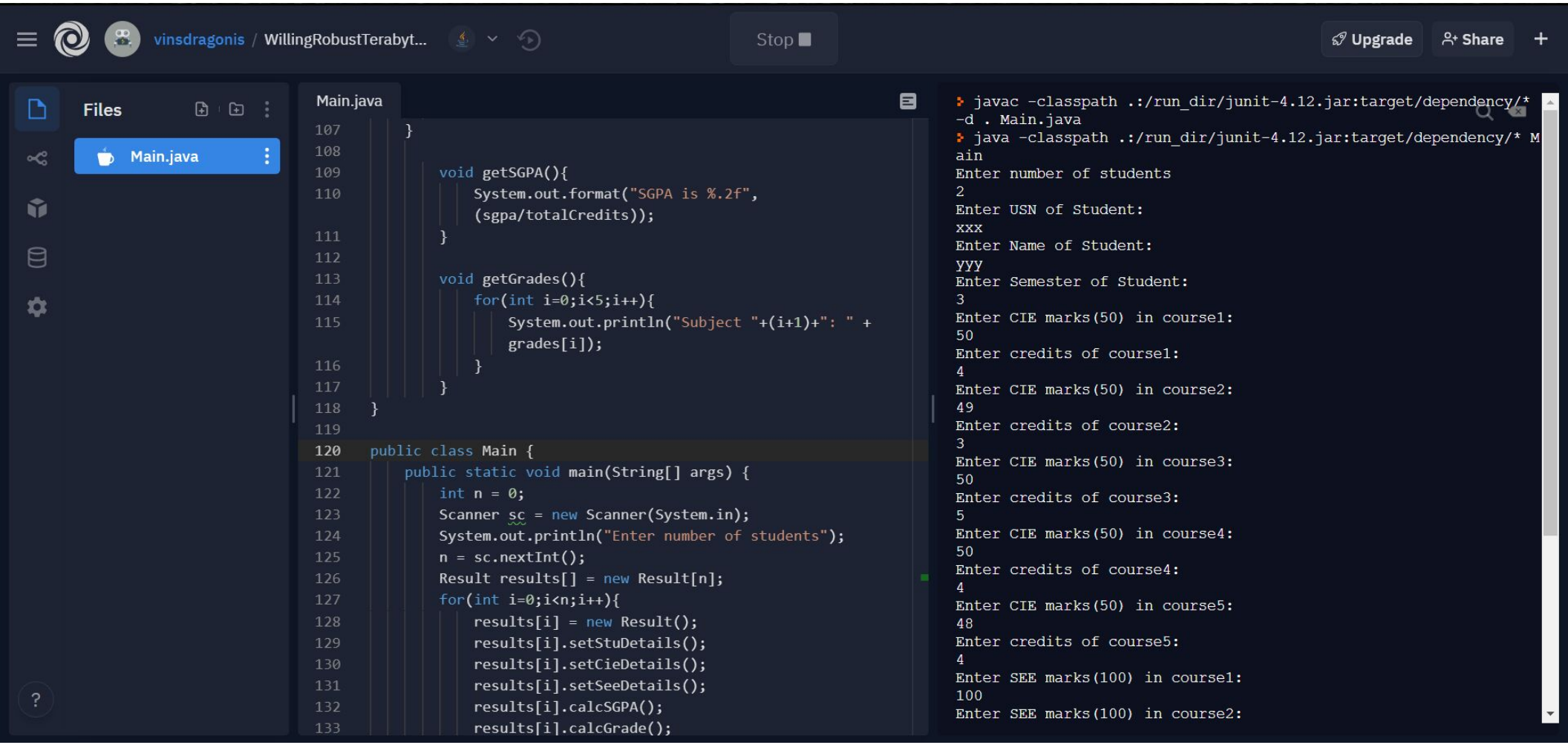
```
    SetData();
}

@override
public String getName() {
    return this.name;
}

@override
public void setName() {
    String name = S.nextLine();
    this.name = name;
}

@override
public int getMatches() {
    return matchesplayed;
}

@Override
public void setMatches() {
    int matches = S.nextInt();
    this.matchesplayed = matches;
}
public double getRunsScored() {
    return runsScored;
}

public void SetRunsScored() {
    double rune = S.nextDouble();
    this.runsScored = rune;
}

public void SetData() {
```

```
System.out.println("Enter batsman name.");
SetName();
System.out.println("Enter total runs scored
  from all matches.");
SetRunsScored();
}

@override
public double getAvgScore() {
  return avg;
}

@override
public void SetAvgScore() {
  this.avg = getRunsScored() / getMatches();
}

@Override
public void printData() {
SetAvgScore();

System.out.println("Name = " + this.getName() +"
  \n Total runsScored = " + this.getRunsScored() +
  \n Average score = " + this.getAvgScore());
}
}
class Bowler extends player {
  private double runsGiven;

public Bowler() {
  setData();
}

@override
```

```
public String getName() {
  return this.name;
}

@override
public void SetName() {
  String name = S.nextLine();
  this.name = name;
}

@Overyride
public int getMatches() {
  return matchesPlayed;
}

@override
public void SetMatches() {
  int matches = S.nextInt();
  this.matchesplayed = matches;
}

@overrride
public double getAvgScore() {
  return avg;
}

Public double getRunsGiven() {
  return this.runsGiven;
}

public void SetRunsGiven() {
  double runs = S.nextDouble();
  this.runsGiven = runs;
}
```

```
public void setData() {
 System.out.println("Enter bowler name.");
 setName();
System.out.println("Enter total number of
                matches.");
 setMatches();
System.out.println("Enter total runs given
      from all matches.");
 setRunsGiven();
}

 @Override
 public void SetAvgScore() {
   this.avg = getRunsGiven() / getMatches();
 }

 @Override
  public void printData() {
   SetAvgScore();
 System.out.println("Name = " + this.getName() +
   " \n Total runs given = " + this.getRunsGiven()
   + \n Average Score = " + this.getAvgScore());
  }
}

class playerAvg {
  public static void main(String[] args) {
   int m,n, i;
  Scanner s = new Scanner(System.in);

 System.out.println("Enter number of
          batsman");
 m = s.nextInt();
  player bat[] = new Batsman[m];
```

```
System.out.println("Enter number of bowl
n = s.nextInt();
player ball[] = new Bowler[n];

for (i=0; i<m; i++) {
  bat[i] = new Batsman();
  bat[i].printData();
}

  for (i=0; i<n; i++) {
    ball[i] = new Bowler();
    ball[i] = printData();
  }
 }
}
```


vinsdragonis / WillingRobustTerabyt...
Stop
Upgrade
Share
Files
Main.java
Main.java
}
void getSGPA(){
System.out.format("SGPA is %.2f",
(sgpa/totalCredits));
}
void getGrades(){
for(int i=0;i<5;i++){
System.out.println("Subject "+(i+1)+": " +
grades[i]);
}
}
}
public class Main {
public static void main(String[] args) {
int n = 0;
Scanner sc = new Scanner(System.in);
System.out.println("Enter number of students");
n = sc.nextInt();
Result results[] = new Result[n];
for(int i=0;i<n;i++){
results[i] = new Result();
results[i].setStuDetails();
results[i].setCieDetails();
results[i].setSeeDetails();
results[i].calcSGPA();
results[i].calcGrade();
javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/*
-d . Main.java
java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* M
ain
Enter number of students
2
Enter USN of Student:
xxx
Enter Name of Student:
yyy
Enter Semester of Student:
3
Enter CIE marks(50) in course1:
50
Enter credits of course1:
4
Enter CIE marks(50) in course2:
49
Enter credits of course2:
3
Enter CIE marks(50) in course3:
50
Enter credits of course3:
5
Enter CIE marks(50) in course4:
50
Enter credits of course4:
4
Enter CIE marks(50) in course5:
48
Enter credits of course5:
4
Enter SEE marks(100) in course1:
100
Enter SEE marks(100) in course2:

3/11/2020

Develop a Java program to create an abstract class named Shape that contains two integers & an empty method named printArea(). Provide three classes named Rectangle, Triangle & circle such that each one of these classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.*;

abstract class Shape {
  int x;

public Shape(int x) {
  this.x = x;
}
public astract void printArea();
}
class Rectangle extends Shape {
  private int y;

  public Rectangle() {
    this(1, 1);
  }

  public Rectangle(int x, int y) {
    super(x);
    this.y = y;
  }
  public int getwidth() {
    return super.x
  }
```

```
public Void SetWidth() {
Scanner s= new Scanner (System.in);
int width;

System.out.println ("Enter width.");
width = s.nextInt();
Super.x = width;
}

public int getHeight() {
 return this.y;
}

public Void SetHeight() {
 Scanner s= new Scanner(System.in);
 int height;

 System.out.println ("Enter height.");
 height = s.nextInt();
 this.y = height;
}

@Override
public Void printArea() {
 SetWidth().
 SetHeight();
 System.out.println("Area: " + (getWidth() *
                    getHeight()));
 }
}

class circle extends Shape {
 private int y;
```

```
public Circle() {
    this(1);
}

public Circle(int x) {
  super(x);
}

public int getRading() {
  return super.x;
}

public void SetRading() {
  Scanner s = new Scanner(System.in);
  int rading;

  System.out.println("Enter rading.");
  rading = s.nextInt();
  super.x = rading;
}

@Override
public void printArea() {
  SetRading();
  System.out.println("Area= " + (Math.PI *
          Math.pow(getRading(), 2)));
  }
}

class Triangle extends Shape {
private int y;

public Triangle() {
  this(1, 1);
}
```

```
public Triangle(int x, int y) {
  Super (x);
  this.y = y;
}

public int getBase() {
  return Super.x;
}

public void SetBase() {
  Scanner S = new Scanner(System.in);
  int base;

System.out.println("Enter base);
base = S.nextInt();
Super.x = base;
}

public int getHeight() {
   return this.y;
}

public Void SetHeight() {
  Scanner S = new Scanner(System.in);
  int height;

System.Out.println("Enter height.");
height = S.nextInt();
this.y = height;
  }

@Override
public void printArea() {
   SetBase();
```

```
        setHeight();
        System.out.println("Area = " +
            ((0.5) * getBase() * getHeight()));
    }
}

class ShapeArea{
    public static void main(String[] args){
        Shape rect = new Rectangle();
        Shape circle = new Circle();
        Shape Triangle = new Triangle();
        int opt = 0;
        Scanner s= new Scanner(System.in);

        while(opt != 4){
        System.out.println("Your option are:
            \n1. Rectangle \n2. circle \n3.
            Triangle.");
        opt = s.nextInt();

    If (op == 1)
        rect.printArea();
    else if (opt == 2)
        circle.printArea();
    else if (opt == 3)
        triangle.printArea();
    elseif (opt == 4)
        System.out.println("Terminating
                    Session.");

    else
        System.out.println("Invalid value
                    passed.");

    }
}
```

2) Develop a Java program to create a class Bank that maintains two kinds of accounts for its customers, one called Savings account & the other current account. The Savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holder should also maintain minimum balance & if it falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number & type of account. From this derive the classes Curr-acct & Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: Accept deposit from customer & update the balance.

- Display the Balance
- Compute & deposit interest
- permit withdrawal & update the balance
- ~~Display~~ the ~~balance~~ check for minimum balance
- ~~compute & deposit interest~~ Impose penalty if necessary & update the balance.

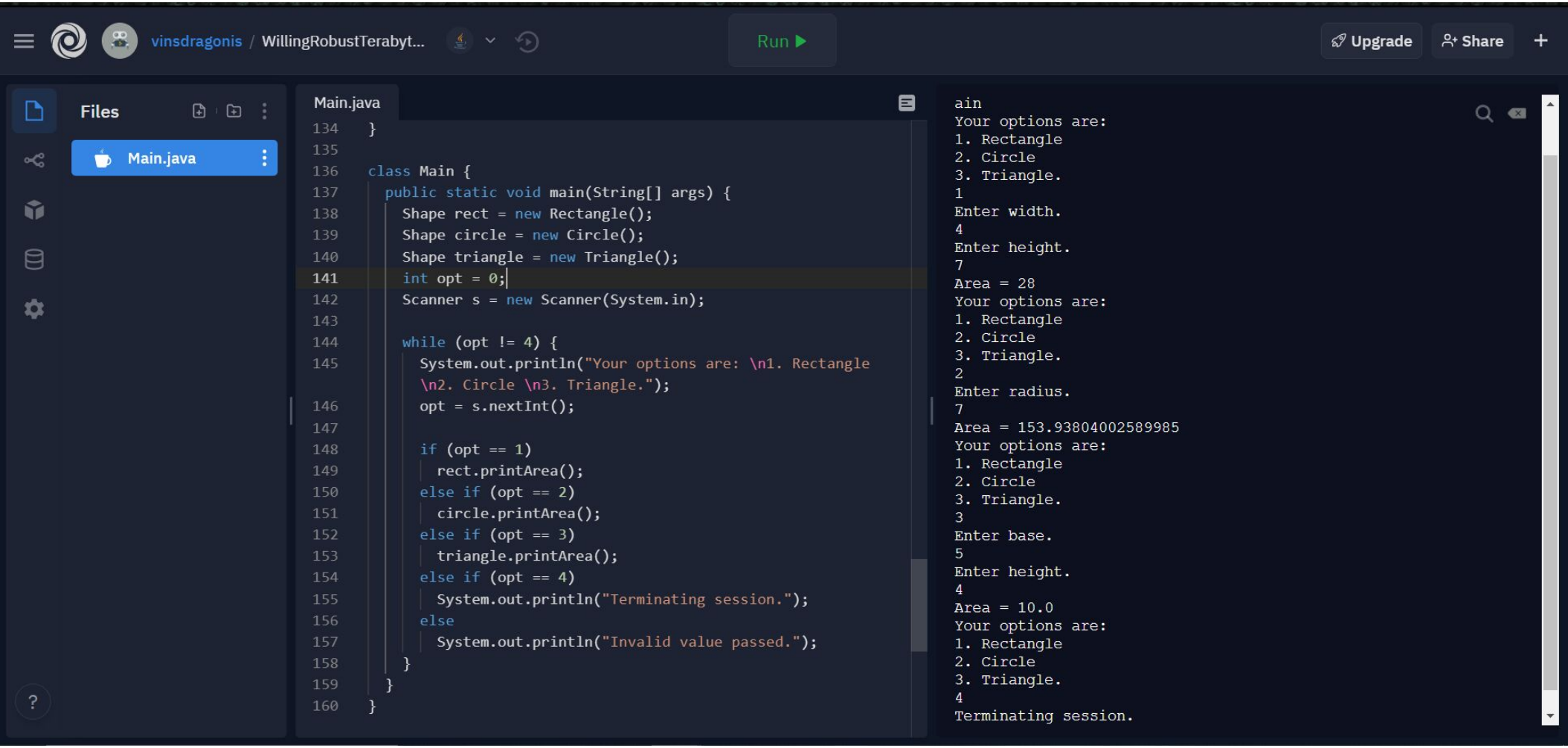
```
import java.util.Scanner;
import java.lang.Math;

class Account
{ String name, type, accno;
  double balance;

  Void deposit()
```

```
{ Scanner get = new Scanner(System.in);
  double depo;
System.out.println ("Enter the deposit:");
   depo = get.nextdouble();
   balance = balance + depo;
}

Void withdraw()
{Scanner get = new Scanner(System.in);
 double withdraw;
   System.out.println ("Enter the amount
    to withdraw : (<"+balance+")");
withdraw = get.nextDouble();
balance = balance + withdraw;
System.out.println("Balance: " + balance);
}
Void create()
{Scanner get = new Scanner(System.in);
 System.out.println("Name:");
   name = get.next();
 type = "Current";
    System.out.println("Account No:");
    accno = get.next();
    System.out.println("Balance:");
    balance = get.nextDouble();
 }
 Void check()
 { System.out.println("\n Minimum Balance: "
                        + 5000);

if (balance < 5000)
   {System.out.println("penalty is imposed
   please deposit minimum" + (5000 - balance
   + 200) + "Rs \n Rs 200 Service charge");
   deposit();
```

```
        balance= balance-200;
      }
      else
      { System.out.println("Balance: "+ balance +
        "Safe");}
    }
}

class Sav_acct extends Account
{ double intr=7;
  boolean cheque=false;
  void disp_blnc()
  {System.out.println("Balance: "+balance);
  }
  void create()
  { Scanner get=new Scanner(Sytem.in);
    System.out.println("name: ");
     name= get.next();
    type= "Savings";
    System.out.println("Balance: ");
    balance= get.nextDouble();
  }

Void calcint()
{ double interest;
  Scanner get= new Scanner(System.in);
  System.out.println("Enter time:");
   int time;
  time= get.nextInt();
  interest= balance * Math.pow(1+ intr/100, time)-balance;
  System.out.println("Interest: "+interest);
  balance= balance+interest;
  System.out.println("Balance: "+balance);
}
}
```

```
class Bank
{ public static void main(String args[])
{ Scanner get= new Scanner(System.in);
  String type;
  Sav_acct accs= new Sav_acct();
  Curr_acct accr= new Curr_acct();
  System.out.println("Enter type of account:
      (Current / Saving)");
  type= get.next();
  if (type.equals("Savings"))
  accs.create();
  elseif (type.equals("Current"))
  accr.Create();

  int ch;
  do
{System.out.println("\n1. Deposit \n2. Display
  Balance \n3. Deposit Interest \n4. Withdrawal
  \n5. Check \n6. Cheque Book (under development)
  \n 7. Exit");
  ch= get.nextInt();
  Switch(ch)
  { case 1: if (type.equals("Savings"))
      accs.deposit();
      else
      accr.deposit();
      break;
    case 2: If (type.equals("Saving"))
      accs.displbnc();
      else
      accr.displbnc();
      break;
    case 3: if (type.equals("Savings"))
      accs.calcint();
```

```
else
  System.out.println("This account doesn
   have this provision");
   break;
case 4: if(type.equals("Savings"))
  acc.withdraw();
 else
    acc.withdraw();
    break;
case 5: if (type.equals("Savings"))
  System.out.println("This account doesnot
        have this provision");
 else
    acc.check();
     break;
case 6: if (type.equals("Savings"))
  System.out.println("This account doesnot
     have this provision");
 else
  System.out.println("This account doe
     have this provision");
     break;
  default: if (ch!=7)
     System.out.println("Enter valid option");
  }
 } while (ch!=7);
}
}
```


vinsdragonis / WillingRobustTerabyt...
Run
Upgrade
Share
Files
Main.java
Main.java
}
class Main {
public static void main(String[] args) {
Shape rect = new Rectangle();
Shape circle = new Circle();
Shape triangle = new Triangle();
int opt = 0;
Scanner s = new Scanner(System.in);
while (opt != 4) {
System.out.println("Your options are: \n1. Rectangle \n2. Circle \n3. Triangle.");
opt = s.nextInt();
if (opt == 1)
rect.printArea();
else if (opt == 2)
circle.printArea();
else if (opt == 3)
triangle.printArea();
else if (opt == 4)
System.out.println("Terminating session.");
else
System.out.println("Invalid value passed.");
}
}
}
ain
Your options are:
1. Rectangle
2. Circle
3. Triangle.
1
Enter width.
4
Enter height.
7
Area = 28
Your options are:
1. Rectangle
2. Circle
3. Triangle.
2
Enter radius.
7
Area = 153.93804002589985
Your options are:
1. Rectangle
2. Circle
3. Triangle.
3
Enter base.
5
Enter height.
4
Area = 10.0
Your options are:
1. Rectangle
2. Circle
3. Triangle.
4
Terminating session.

Week 9

Qu: Create a package CIE which has two classes Student & internals. The class student has members like Usn, name, Sem. The class Internals has an array that stores the internal marks scored in five courses of the current Semester of the student. Create another package SEE which has the class ~~external~~ SEE marks scored in five courses of the current semeste of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

prg:

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

public class Student {
 private String USN;
 private String name;
 private int Sem;
 Scanner s= new Scanner(System.in);

public Student() {
  this("unknown", "unknown", 3);
}

public student(String USN, String name,
               int sem) {
  this.USN = USN;
  this.name= name;
  this.Sem = Sem;
}
```

```
public String getUSN() {
 return USN;
}

public void setUSN (String USN) {
  this.USN= USN;
}

public String getName() {
   return name;
}

public void setName(String name) {
    this.name = name;
}

public int getSem() {
  return sem;
}

public void getData() {
  String USN, name;
  int sem;

  System.out.println("Enter USN");
  USN = S.nextLine();
  setUSN (USN);

  System.out.println("Enter Student name");
  name = S.nextLine();
  SetName (name);

  System.out.println("Enter sem");
   sem = S.nextInt();
   setSem (sem);
}
```

```
public Void printData() {
    System.out.println("USN: " + getUSN());
    System.out.println("Name: " + getName());
    System.out.println("Sem: " + getSem());
  }
}
```

```
package CIE;

public class Internals extends Student {
  private int[] marks;

  public Internals() {
    super();
    marks = new int[5];
  }

  public int[] getMarks() {
    return marks;
  }

  public void setMarks(int[] marks) {
    this.marks = marks;
  }

  public void getCIEMarks() {
    for (int i=0; i<5; i++) {
    System.out.println("Enter CIE marks in Subject" 
      + (i+1));
    System.out.print(">>> ");
    marks[i] = s.nextInt();

    if (marks[i] > 50)
      marks[i] = 50;

    if (marks[i] < 0)
      marks[i] = 0;
    }

    setMarks(marks);
  }
```

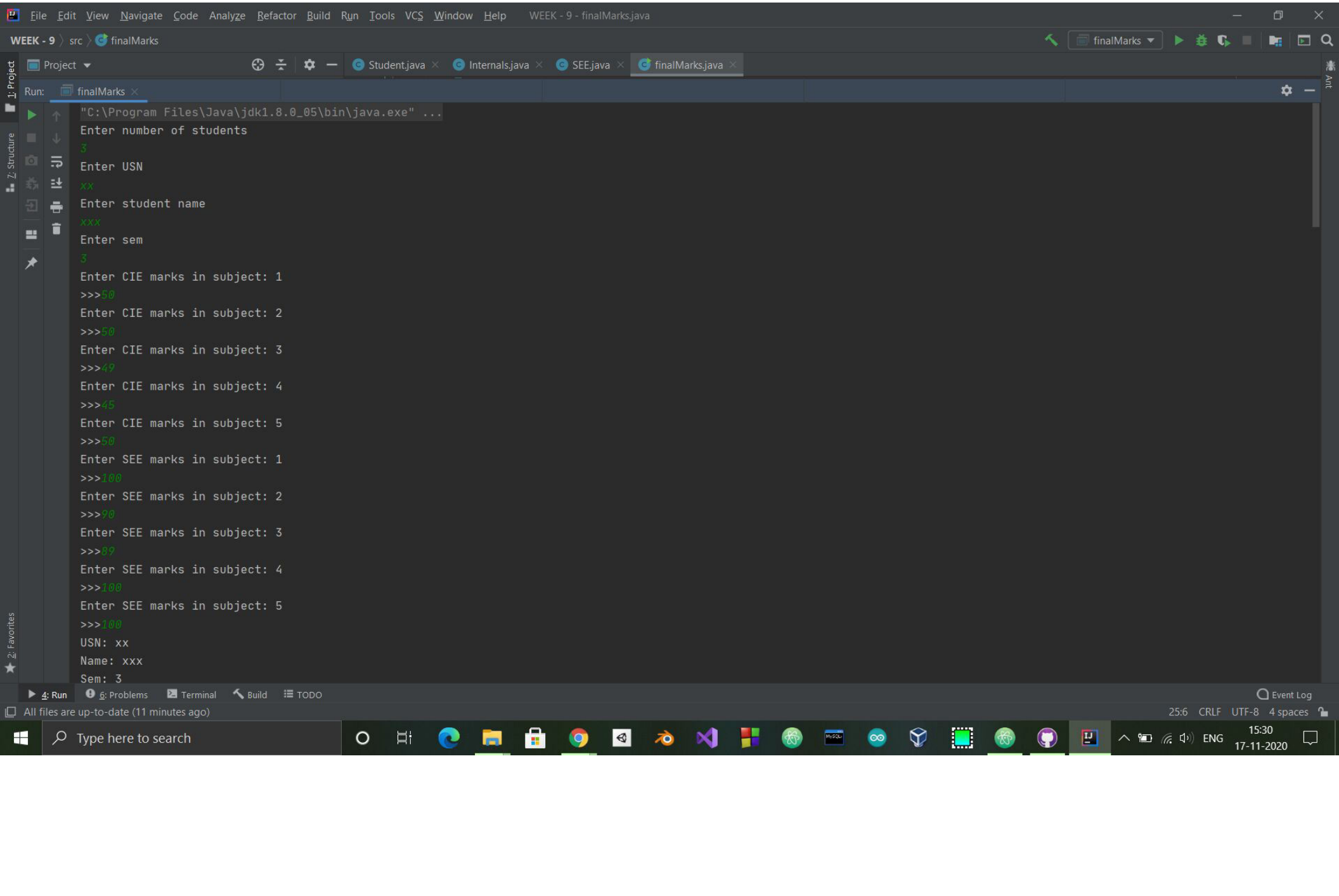
```
public void printCIEMarks() {
    for (int i=0; i<5; i++) {
        System.out.println("CIE marks in
        Subject " + (i+1) + ":" + marks[i]);
    }
}
}
```

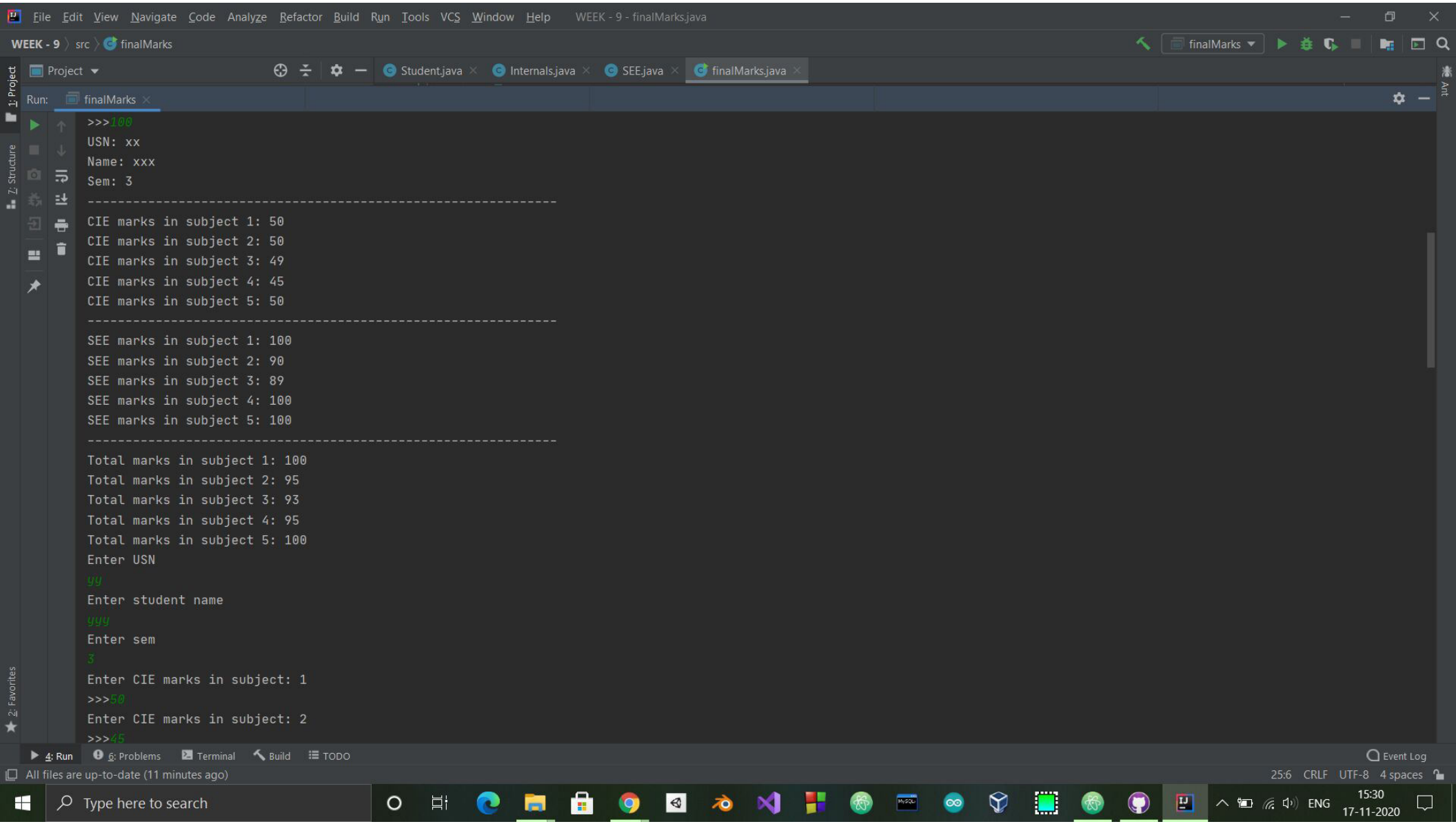
```
package SEE;
import java.lang.*;
import java.util.Scanner;

public class SEE extends CIE.Student {
 private int[] marks;
 Scanner s = new Scanner(System.in);

 public SEE() {
   super();
   marks = new int[5];
 }

 public int[] getMarks() {
   return marks;
 }

 public void SetMarks(int[] marks) {
   this.marks = marks;
 }

 public void getSEEMarks() {
  for(int i=0; i<5; i++) {
  System.out.println("Enter SEE marks in
     Subject : " + (i+1));
  System.out.println(">>>");
  marks[i] = s.nextInt();

  if(marks[i] >100)
     marks[i] = 100;

  if(marks[i] < 0)
     marks[i] = 0;
 }
```

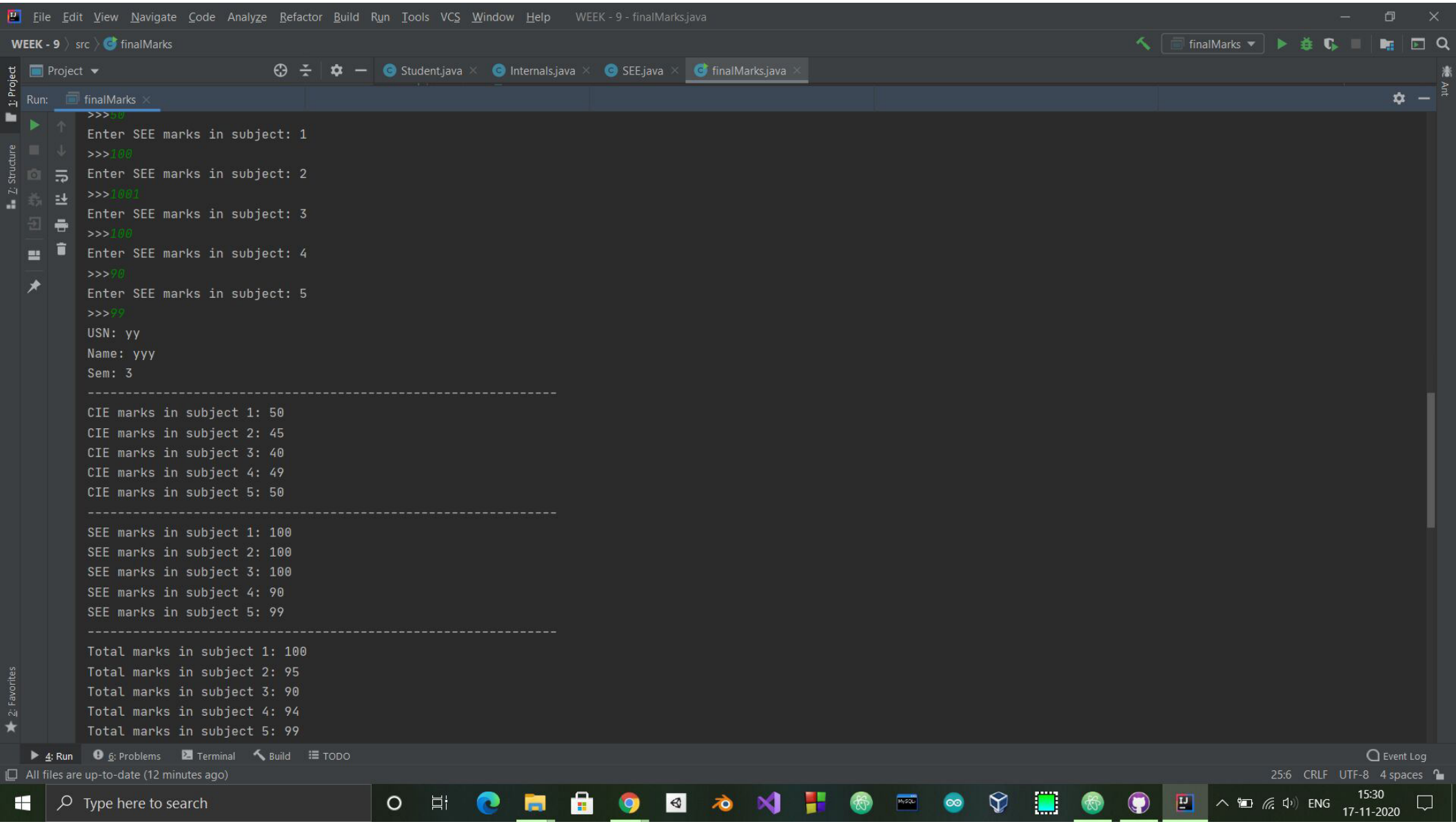
```
        SetMarks(marks);
    }

    public void printSEEMarks() {
        for(int i=0; i<5; i++) {
            System.out.println("SEE marks in subject"
                + (i+1) + ":" + marks[i]);
        }
    }
}
```

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

public class finalmarks {
  public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int n, count;

    System.out.println("Enter number of students");
    n = s.nextInt();

    for (count = 0; count<n; count++) {
      CIE.Internals cie = new Internal();
      SEE see = new SEE();

      cie.getData();
      cie.getCIEMarks();
      see.getSEEMarks();

      int[] CIEMarks = cie.getMarks();
      int[] SEEMarks = see.getMarks();
      int[] totalMarks = new int[5];

      for (int i=0; i<5; i++) {
        totalMarks[i] = CIEMarks[i] + (SEEMarks[i]/2);
      }

      cie.printData();
      System.out.println("----------");
      cie.printCIEMarks();
      System.out.println("----------");
      see.printSEEMarks();
      System.out.println("----------");
```

```
        for (int i=0; i<5 ; i++) {
            System.out.println ("Total marks in
            subject " + " + totalMarks [i]);
        }
    }
}
```


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
WEEK - 9 - finalMarks.java
WEEK - 9 〉 src 〉 finalMarks
finalMarks
Project
Student.java
Internals.java
SEE.java
finalMarks.java
Run: finalMarks
"C:\Program Files\Java\jdk1.8.0_05\bin\java.exe" ...
Enter number of students
3
Enter USN
xx
Enter student name
xxx
Enter sem
3
Enter CIE marks in subject: 1
>>>50
Enter CIE marks in subject: 2
>>>50
Enter CIE marks in subject: 3
>>>49
Enter CIE marks in subject: 4
>>>45
Enter CIE marks in subject: 5
>>>50
Enter SEE marks in subject: 1
>>>100
Enter SEE marks in subject: 2
>>>90
Enter SEE marks in subject: 3
>>>89
Enter SEE marks in subject: 4
>>>100
Enter SEE marks in subject: 5
>>>100
USN: xx
Name: xxx
Sem: 3
4: Run
6: Problems
Terminal
Build
TODO
Event Log
All files are up-to-date (11 minutes ago)
25:6 CRLF UTF-8 4 spaces
Type here to search
ENG
15:30
17-11-2020

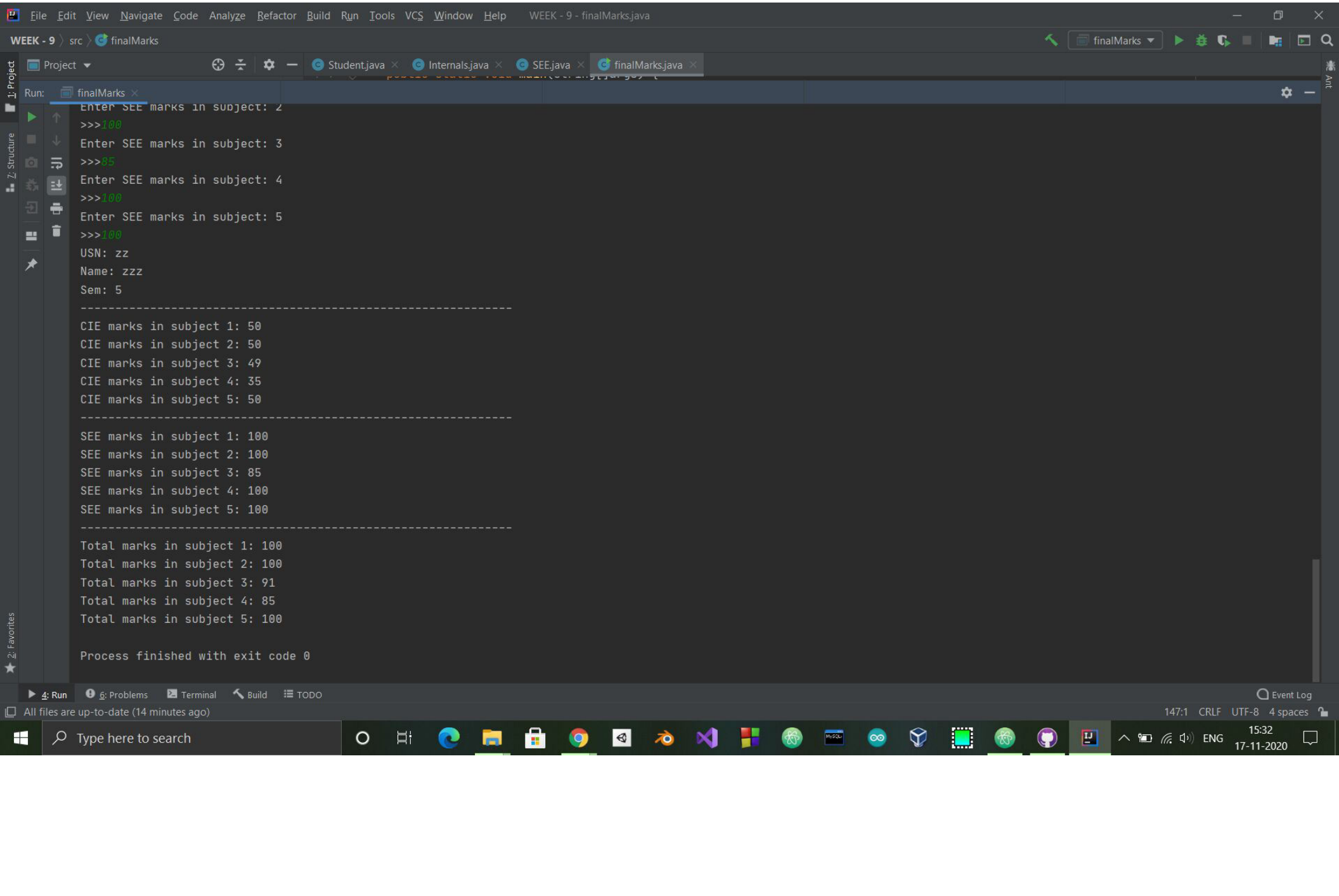

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
WEEK - 9 - finalMarks.java
WEEK - 9 〉 src 〉 finalMarks
finalMarks
Project
Student.java
Internals.java
SEE.java
finalMarks.java
Run: finalMarks
>>>100
USN: xx
Name: xxx
Sem: 3
CIE marks in subject 1: 50
CIE marks in subject 2: 50
CIE marks in subject 3: 49
CIE marks in subject 4: 45
CIE marks in subject 5: 50
SEE marks in subject 1: 100
SEE marks in subject 2: 90
SEE marks in subject 3: 89
SEE marks in subject 4: 100
SEE marks in subject 5: 100
Total marks in subject 1: 100
Total marks in subject 2: 95
Total marks in subject 3: 93
Total marks in subject 4: 95
Total marks in subject 5: 100
Enter USN
yy
Enter student name
yyy
Enter sem
3
Enter CIE marks in subject: 1
>>>50
Enter CIE marks in subject: 2
>>>45
4: Run
6: Problems
Terminal
Build
TODO
Event Log
All files are up-to-date (11 minutes ago)
25:6 CRLF UTF-8 4 spaces
Type here to search
ENG
15:30
17-11-2020


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
WEEK - 9 - finalMarks.java
WEEK - 9 〉 src 〉 finalMarks
finalMarks
Project
Student.java
Internals.java
SEE.java
finalMarks.java
Run: finalMarks
>>>50
Enter SEE marks in subject: 1
>>>100
Enter SEE marks in subject: 2
>>>1001
Enter SEE marks in subject: 3
>>>100
Enter SEE marks in subject: 4
>>>90
Enter SEE marks in subject: 5
>>>99
USN: yy
Name: yyy
Sem: 3

CIE marks in subject 1: 50
CIE marks in subject 2: 45
CIE marks in subject 3: 40
CIE marks in subject 4: 49
CIE marks in subject 5: 50

SEE marks in subject 1: 100
SEE marks in subject 2: 100
SEE marks in subject 3: 100
SEE marks in subject 4: 90
SEE marks in subject 5: 99

Total marks in subject 1: 100
Total marks in subject 2: 95
Total marks in subject 3: 90
Total marks in subject 4: 94
Total marks in subject 5: 99
4: Run
6: Problems
Terminal
Build
TODO
Event Log
All files are up-to-date (12 minutes ago)
25:6 CRLF UTF-8 4 spaces
Type here to search
ENG
15:30
17-11-2020


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
WEEK - 9 - finalMarks.java
WEEK - 9 〉 src 〉 finalMarks
finalMarks
Project
Student.java
Internals.java
SEE.java
finalMarks.java
Run: finalMarks
Enter SEE marks in subject: 2
>>>100
Enter SEE marks in subject: 3
>>>85
Enter SEE marks in subject: 4
>>>100
Enter SEE marks in subject: 5
>>>100
USN: zz
Name: zzz
Sem: 5

CIE marks in subject 1: 50
CIE marks in subject 2: 50
CIE marks in subject 3: 49
CIE marks in subject 4: 35
CIE marks in subject 5: 50

SEE marks in subject 1: 100
SEE marks in subject 2: 100
SEE marks in subject 3: 85
SEE marks in subject 4: 100
SEE marks in subject 5: 100

Total marks in subject 1: 100
Total marks in subject 2: 100
Total marks in subject 3: 91
Total marks in subject 4: 85
Total marks in subject 5: 100
Process finished with exit code 0
4: Run
6: Problems
Terminal
Build
TODO
Event Log
All files are up-to-date (14 minutes ago)
147:1 CRLF UTF-8 4 spaces
Type here to search
ENG
15:32
17-11-2020

Write a program to demonstrate generics with multiple Object parameter

```
class Test <X, Y, Z, W) {
  X projects;
  Y name;
  Z exp;
  W languages;

Test (X projects, Y name, Z exp,
        W languages) {

this.projects = projects;
this.name = name;
this.exp = exp;
this.languages = languages;
}

public X getprojects () {
  return this.projects;
}

public Y getName() {
return this.name;
}

public Z getExp () {
  return this.exp;
}

public getExp() {
  return this.languages;
}
```

```
public void print() {
    System.out.println("Number of project: " + getproject() + "\n Developer Name
    " + getName() + "\n Industry experience: " +
    getExp() + "\n known programming language
    : " + getLang());
  }
}

Class Generics {
  public static void main (String[] args) {
    Test < Integer, String, Integer, Integer >
    t= new Test < Integer, String, Integer,
      Integer > (10, "Vineeth", 11, 12);
    t.print();
  }
}
```

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & derived class called "Son" which extends the base class.

```
import java.lang.*;
import java.util.Scanner;

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

class Father
 {
  private int age;

  Father(int age) {
    this.age = age;
  }

  public int getAge() {
     return this.age;
  }
}

class Son extends Father
 {
   private int sAge;

   Son(int fAge, int sAge) {
     super(fAge);
     this.sAge = sAge;
   }
```

```
public int getSAge() {
    return this.sAge;
}

public void disp() throws WrongAge {
 if (super.getAge() <= getsAge() ||
 super.getAge() < 0 || getsAge() < 0)
  throw new WrongAge("please enter
    a vaid age value.");
  else
   System.out.println("Father's age: " +
    super.getAge() + "\n son's age: " +
      this.getSAge());
  }
}
```

```
class ExceptionalHanding {
  public static void main(String args[]){
   int dad = 1, son = 1;
   Scanner s = new Scanner(System.in);

  try {
   dad = s.nextInt();
   son = s.nextInt();
   Son S1 = new Son(dad, son);

   S1.disp();
  }
  catch (WrongAge e) {
    e.printStackTrace();
  }
 }
}
```

```java
/*Write a program to demonstrate generics with multiple object parameters.*/

class Test<X, Y, Z, W> {
  X projects;
  Y name;
  Z exp;
  W languages;

  Test(X projects, Y name, Z exp, W languages) {
    this.projects = projects;
    this.name = name;
    this.exp = exp;
    this.languages = languages;
  }

  public X getProjects() {
    return this.projects;
  }

  public Y getName() {
    return this.name;
  }

  public Z getExp() {
    return this.exp;
  }

  public W getLang() {
    return this.languages;
```

Generics.java

Compile Messages | jGRASP Messages | Run I/O | Interactions

End | Clear | Help

```
----jGRASP exec: java Generics
Number of Projects: 10
Developer Name: Vineeth
Industry Expeience: 11
Known programming languages: 12
```

Line:1 Col:1 Code:47 Top:1 OVS BLK

Week - 11

Write a program which creates two threads one thread "BMS College of Engineering" once every 10 seconds & another displaying "CSE" once every two seconds.

```
class NewThread implements Runnable {
    private String name;
    private int interval;
    private Thread t;

NewThread (String threadname, int interval) {
    this.name = threadname;
    this.interval = interval;
    t = new Thread (this, name);
    t.start ();
}

Public void run() {
    int i=1, n=5;
    try {
        while (i<=n) {
        System.out.println ("Thread:" +
                            this.name);
        i++;
        Thread.sleep (this.interval);
        }
    }
    catch (InterruptedException e) {
        System.out.println (name + "Interrupted");
    }
}
}
```

```
class MultiThread {
    public static void main (String args[]) {
        new NewThread ("BMSCE", 10000);
        new NewThread ("CSE", 2000);
    }
}
```

```
 ----jGRASP exec: java MultiThread
Thread: CSE
Thread: BMSCE
Thread: CSE
Thread: CSE
Thread: CSE
Thread: CSE
Thread: BMSCE
Thread: BMSCE
Thread: BMSCE
Thread: BMSCE

 ----jGRASP: operation complete.
```

5/12/2020

WEEK - 12.

classmate
Date
Page

Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields Num1 & Num2. The division of Num1 & Num2 is displayed in the result field when the divided button is clicked. If Num1 & Num2 were not integers, the program would throw NumberFormat - exception. If Num2 were Zero, the program would throw an arithmetic exception. Display exception in a message dialog box.

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

class dialogERR {
   JFrame frame;

dialogERR() {
  frame = new JFrame();
  JOptionPane.showMessageDialog(frame, "cannot
   divide by non-integer values");
}
}

public class AWT_Division extends Frame
   implements ActionListener {
  Textfield tf1, tf2;
  Label l;
  Button b;
```

```
Main() {
    tf1 = new TextField();
    tf1.setBounds(75, 50, 200, 25);

    tf2 = new TextField();
    tf2.setBounds(75, 100, 200, 25);

    l = new Label();

    b = new Button("Divide");
    b.setBounds(125, 200, 100, 50);
    b.addActionListener(this);

    add(b);
    add(tf1);
    add(tf2);
    setSize(350, 350);

    setLayout(null);
    setVisible(true);
}

public void actionPerformed(ActionEvent e){
  try {
        String n1 = tf1.getText();
        String n2 = tf2.getText();
        int width, x;
        int q = Integer.parseInt(n1) /
                Integer.parseInt(n2);

    If ((int)Math.floor(Math.log10(Math.
       abs(q)) + 1) == 1)
     width = 75;
    elseif ((int).floor(Math.log10(Math.abs(q)
                                    + 1) == 2)
```

```
width = 90;
else
  width = 110;

x = (350 - width) / 2;
l.setBounds(x, 150, width, 50);
l.setText("Quotient: " + q);
add(l);
}

catch (NumberFormatException ne) {
 new dialogERR2();
}

catch (ArithmeticException ze) {
l.setBounds(125, 150, 100, 50);
l.setText("Cannot divide by zero");
add(l);
}

catch (Exception ex) {
 System.out.println(ex);
}
}

public static void main(String [] args){
 new Main();
}

}
```

```
      add(l);
    }
    catch(NumberFormatException ne) {
      new dialogERR();
    }
    catch(ArithmeticException ze) {
      l.setBounds(125, 150, 100, 50);
      l.setText("Cannot divide by zero");
      add(l);
    }
    catch(Exception ex) {
      System.out.println(ex);
    }
  }

  public
    new
  }
}
```