EAST WEST UNIVERSITY

Coures : CSE110 , Object Oriented Programming

Assignment : Lab - 6

Submitted By

Name : Ali Haidar

ID : 2022-1-60-193

Submitted To

Mahamudul Hasan

(Senior Lecturer)

Department of Computer Science & Engineering

Date : 27 – 07 – 2023

Problem – 1 (a) : Define a class Line that has four private instance variables: (x1, y1) and (x2, y2), representing the value of the x-coordinate and y-coordinate of the first point and second point on the line, respectively, in a two-dimensional coordinate system.

The following figure shows a line with these two points. The class Line has the following instance methods.

- findSlope(): the method takes no parameter and returns the slope of the line. The formula for calculating the slope of a line using two points is given below. slope = $y2-y1\ x2-x1$
- toString(): the method returns a String containing the values of (x1, y1) and (x2, y2) on a line.

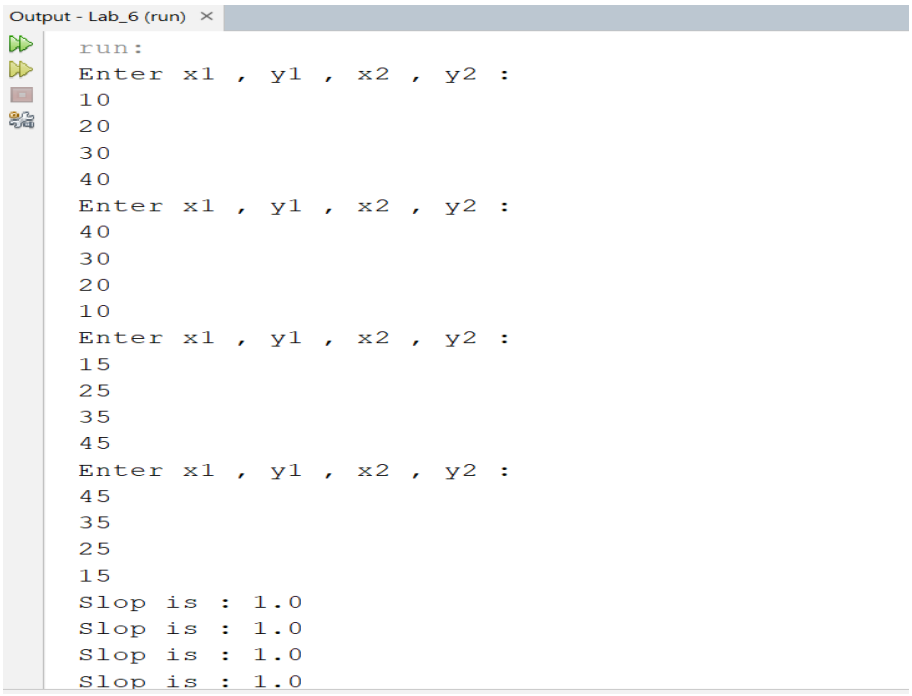
As an example, for a line that has two points (3, 4) and (9, 5), the method may return the following String. Line has points (3, 4) and (9, 5)

Code :

```
package lab_6;

public class Line {

   private int x1;
   private int y1;
   private int x2;
   private int y2;

   public Line() {

   }
   public Line(int x1, int y1, int x2, int y2) {
      this.x1 = x1;
      this.y1 = y1;
      this.x2 = x2;
      this.y2 = y2;
   }

   public void findSlop() {
      double slp = ((y2 - y1) / (x2 - x1));
      System.out.println("Slop is : " + slp);
   }
}
```

```
package lab_6;
import java.util.Scanner;

public class Test {

   public static void main(String[] args) {
      int x1, x2, y1, y2;

      Line arr[] = new Line[4];
      Scanner x = new Scanner(System.in);
      for (int i = 0; i < 4; i++) {
         System.out.println("Enter x1 , y1 , x2 , y2 : ");
         x1 = x.nextInt();
         y1 = x.nextInt();
         x2 = x.nextInt();
         y2 = x.nextInt();
         arr[i] = new Line(x1, y1, x2, y2);
      }
      for (int i = 0; i < 4; i++) {
         arr[i].findSlop();
      }
   }
}
```

: OutPut :



Problem – 1 (b) : Use appropriate datatype. Your class definition must also include constructors. Write a Java program that creates an array of Line type objects. The size of the array is 4.

User may input the values of x and y-coordinates of two points on those lines using Scanner class. Your program should also have the following static method. isIntersecting(Line l1, Linel2

Code :

```
package lab_6 ;
class Line2 {
   private double x1, y1, x2, y2;
   public Line2(double x1, double y1, double x2, double y2) {
      this.x1 = x1;
      this.y1 = y1;
      this.x2 = x2;
      this.y2 = y2;
   }
   public double getX1() {
      return x1;
   }
   public double getY1() {
      return y1;
   }
   public double getX2() {
      return x2;
   }
   public double getY2() {
      return y2;
   }
   public static boolean isIntersecting(Line2 l1, Line2 l2) {
      double x1 = l1.getX1();
      double y1 = l1.getY1();
      double x2 = l1.getX2();
      double y2 = l1.getY2();
      double x3 = l2.getX1();
      double y3 = l2.getY1();
      double x4 = l2.getX2();
      double y4 = l2.getY2();

      double slope1 = (y2 - y1) / (x2 - x1);
      double slope2 = (y4 - y3) / (x4 - x3);
      if (slope1 == slope2) {
         return false;
      }
      double intersectionX = (y3 - y1 - slope2 * x3 + slope1 * x1) / (slope1 - slope2);
      if ((intersectionX >= Math.min(x1, x2) && intersectionX <= Math.max(x1, x2))
            && (intersectionX >= Math.min(x3, x4) && intersectionX <= Math.max(x3, x4))) {
         return true;
      }
    return false; } }
```

```
package lab_6 ;
import java.util.Scanner;
public class Test2 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      Line2[] lines = new Line2[4];

      for (int i = 0; i < 4; i++) {
         System.out.println("Enter coordinates for Line " + (i + 1));
         System.out.print("x1: ");
         double x1 = scanner.nextDouble();
         System.out.print("y1: ");
         double y1 = scanner.nextDouble();
         System.out.print("x2: ");
         double x2 = scanner.nextDouble();
         System.out.print("y2: ");
         double y2 = scanner.nextDouble();
         lines[i] = new Line2(x1, y1, x2, y2);
      }

      for (int i = 0; i < lines.length; i++) {
         for (int j = i + 1; j < lines.length; j++) {
            if (Line2.isIntersecting(lines[i], lines[j])) {
               System.out.println("Line " + (i + 1) + " and Line " + (j + 1) + " intersect.");
            }
         } } } }
```

: Output :

```
run:
Enter coordinates for Line 1
x1: 10
y1: 20
x2: 30
y2: 40
Enter coordinates for Line 2
x1: 4030
y1: 30
x2: 20
y2: 10
Enter coordinates for Line 3
x1: 20
y1: 36
x2: 54
y2: 20
Enter coordinates for Line 4
x1: 10
y1: 23
x2: 26
y2: 50
Line 1 and Line 3 intersect.
BUILD SUCCESSFUL (total time: 31 seconds)
```

Problem – 2 : Your job is to define the above-mentioned classes as per the specification mentioned below and then write a Main/Driver class that demonstrates the functionalities of these classes.

Developing a Menu-based Application :

• The initial menu may have the following options:

a. Add
b. Delete
c. Update
d. Print
e. Search

For each of these options, we may provide further options.

Suppose, for 'Add' option, next we may show the following options:

a. Add a Student
b. Add a Course
c. Add a Faculty

For 'Delete' and 'Update', we may provide the same options …. …. …

Code :

```
package lab_6;

public class Student {

   String studentName;
   int studentId;
   double CGPA;

   public Student() {

   }
   public Student(String studentName, int studentId, double CGPA
   ) {
      this.studentName = studentName;
      this.studentId = studentId;
      this.CGPA = CGPA;
   }
   @Override
   public String toString() {
      return "\nName :" + studentName + "\nID : " + studentId + "\nCGPA : " + CGPA;
   }

}
```

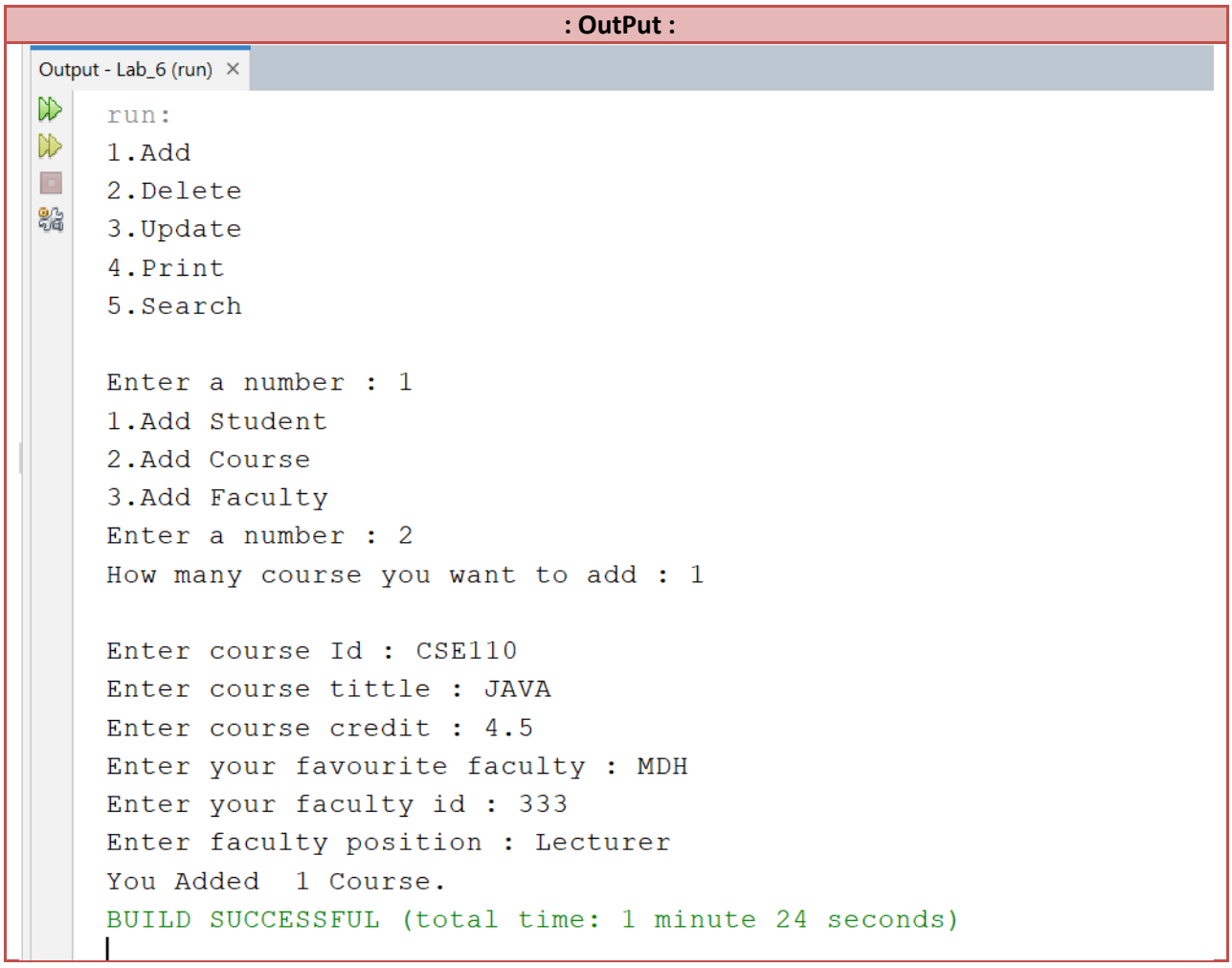
```
package lab_6;

public class Course {

   private String courseID;
   private String courseTittle;
   private double credit;
   private Faculty faculty;

   public Course() {

   }

   public Course(String courseID, String courseTittle, double credit, Faculty faculty) {
      this.courseID = courseID;
      this.courseTittle = courseTittle;
      this.credit = credit;
      this.faculty = faculty;
   }

   @Override
   public String toString() {
      return "\nCourse Id :" + courseID + "\nCourse Tittle : " + courseTittle + "\nCredit : " + credit +
"\nFaculty : " + faculty;
   }

}

class Faculty {

   int facultyId;
   String facultyName;
   String position;
   public Faculty(int facultyId) {
      this.facultyId = facultyId;
   }
   public Faculty(int facultyId, String facultyName, String position) {
      this(facultyId);
      this.facultyName = facultyName;
      this.position = position;
   }

   @Override
   public String toString() {
      return "Faculty{" + "facultyId=" + facultyId + ", facultyName=" + facultyName + ", position=" +
position + '}';
   }
}
```

```
package lab_6;

import java.util.Scanner;

public class Driver {

   public static void main(String[] args) {
      Student[] st = new Student[5];
      Course[] crs = new Course[5];
      Faculty[] fcl = new Faculty[5];

      String name, ctittle, fclty, cid, fname, fposition;
      int id, fid;
      double cgpa, crdt;

      Scanner x = new Scanner(System.in);

      System.out.print("1.Add \n2.Delete  \n3.Update \n4.Print \n5.Search");
      System.out.print("\n\nEnter a number : ");
      int n = x.nextInt();

      switch (n) {
         case 1: {
            System.out.println("1.Add Student \n2.Add Course \n3.Add Faculty ");
            System.out.print("Enter a number : ");
            int nn = x.nextInt();

            if (nn == 1) {
               System.out.print("How many student you want to add : ");
               int stn = x.nextInt();
               for (int i = 0; i < stn; i++) {
                  System.out.print("\nEnter your name : ");
                  name = x.next();
                  System.out.print("Enter your id : ");
                  id = x.nextInt();
                  System.out.print("Enter your cgpa : ");
                  cgpa = x.nextDouble();

                  st[i] = new Student(name, id, cgpa);
                  System.out.println("You Added " + (i + 1) + "Student");
               }

               for (int i = 0; i < stn; i++) {
                  st[i].toString();
               }

            } else if (nn == 2) {
               System.out.print("How many course you want to add : ");
```

```
            int cad = x.nextInt();

            for (int i = 0; i < cad; i++) {
                System.out.print("\nEnter course Id : ");
                cid = x.next();
                System.out.print("Enter course tittle : ");
                ctittle = x.next();
                System.out.print("Enter course credit : ");
                crdt = x.nextDouble();
                System.out.print("Enter your favourite faculty : ");
                fname = x.next();
                System.out.print("Enter your faculty id : ");
                fid = x.nextInt();
                System.out.print("Enter faculty position : ");
                fposition = x.next();

                Faculty f1 = new Faculty(fid, fname, fposition);
                crs[i] = new Course(cid, ctittle, crdt, f1);

                System.out.println("You Added  " + (i + 1) + " Course.");

            }

            for (int i = 0; i < cad; i++) {
                crs[i].toString();
            }
        } else if (nn == 3) {
            System.out.print("How many faculty you want to add : ");
            int fn = x.nextInt();

            for (int i = 0; i < fn; i++) {
                System.out.print("\nEnter faculty name : ");
                fname = x.next();
                System.out.print("Enter faculty ID : ");
                fid = x.nextInt();
                System.out.print("Enter the position : ");
                fposition = x.next();

                fcl[i] = new Faculty(fid, fname, fposition);
            }
            System.out.println("\nSuccessfully Added Faculty .");

        }
        break;

    }

    case 2: {
```

```
            System.out.println("1.Remove a student \n2.Delete a course \n3.Remove a faculty");
            System.out.println("Enter a number : ");
            int nn = x.nextInt();

            if (nn == 1) {
               System.out.print("Enter student Id : ");
               id = x.nextInt();

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

               System.out.println("\n\nSuccessfully Removed .");
            } else if (nn == 2) {
               System.out.print("Enter course Id : ");
               cid = x.next();

               System.out.print("Enter tittle : ");
               ctittle = x.next();

               System.out.println("\n\nSuccessfully Course Removed .");
            } else if (nn == 3) {
               System.out.print("Enter faculty Id: ");
               fid = x.nextInt();

               System.out.print("Enter faculty Name : ");
               fname = x.next();

               System.out.println("\n\nSuccessfully Removed .");
            }
         }
         case 3: {
            System.out.print("Enter old ID : ");
            id = x.nextInt();
            System.out.print("Enter new id : ");
            int id2 = x.nextInt();

            System.out.println("The Student List Is Updated .");
         }
      }
   }
}
```

: OutPut :


Output - Lab_6 (run)
run:
1.Add
2.Delete
3.Update
4.Print
5.Search
Enter a number : 1
1.Add Student
2.Add Course
3.Add Faculty
Enter a number : 2
How many course you want to add : 1
Enter course Id : CSE110
Enter course tittle : JAVA
Enter course credit : 4.5
Enter your favourite faculty : MDH
Enter your faculty id : 333
Enter faculty position : Lecturer
You Added 1 Course.
BUILD SUCCESSFUL (total time: 1 minute 24 seconds)

Problem – A : We have only seen instance variables so far. Consider the following class diagram of Book class.

• Here, ISBN, bookTitle and numberOfPages are instance variables. Each object has their own copy of instance variables.

• On the other hand, objects share only one copy of static variable. Like static methods, static variables belong to a class, not to any objects.

• A static variable such as count in the Book class can be used to count the total number of Book type objects. Each time, a Book type object is created; we need to increment the value of the count variable within the constructor.

• A method that returns the value of a static variable is called a static method. I this Book class, getCount() is a static method that returns the value of count.

• Your job is to implement the Book class as per the above-mentioned class diagram

Code :

```
package lab_6;
public class Book {
   private String ISBN;
   private String bookTitle;
   int numberOfPages;
   private static int count = 0;

   public Book(String ISBN, String bookTitle, int numberOfPages) {
      this.ISBN = ISBN;
      this.bookTitle = bookTitle;
      this.numberOfPages = numberOfPages;
      count++;
   }
   public String getISBN() {
      return ISBN;
   }
   public String getBookTitle() {
      return bookTitle;
   }
   public int getNumberOfPages() {
      return numberOfPages;
   }
   public static int getCount() {
      return count;
   }

}
```

```
package lab_6 ;
public class TestA {
   public static void main(String[] args) {
      Book book1 = new Book("1234", "Math", 200);
      Book book2 = new Book("9876", "English", 250);

      int totalBooks = Book.getCount();
      System.out.println("Total number of books created: " + totalBooks);
   }
}
```

: OutPut :