

Java Assignment 3

Submitted by *Rohan Verma*, **1510110508**

Contains all the code of Non Graded Labs of CSD 206 (Monsoon 2015, Shiv Nadar University)

The code is also uploaded to https://github.com/rhnvrn/java_learn and the commit history is available at https://github.com/rhnvrn/java_learn/commits/master

Lab 0

WelcomeJava.java

```
1 public class WelcomeJava{
2     public static void main(String[] args){
3         System.out.println("Welcome");
4     }
5 }
```

WelcomeJava2.java

```
1 import javax.swing.JOptionPane;
2
3 public class WelcomeJava2{
4     public static void main(String[] args){
5         JOptionPane.showMessageDialog(null, "Welcome to Java");
6         JOptionPane.showMessageDialog(null, "Welcome to Java 3",
7             "GUI Welcome", 3);
8         JOptionPane.showMessageDialog(null, "Welcome to Java 3",
9             "GUI Welcome", 2);
10        JOptionPane.showMessageDialog(null, "Welcome to Java 3",
11            "GUI Welcome", JOptionPane.QUESTION_MESSAGE);
12    }
13 }
```

ComputeArea.java

```
1 import javax.swing.JOptionPane;
2
3
4 public class ComputeArea{
5     public static void main(String[] args){
6         float radius = 0;
7         double area;
```

```

8
9     area = radius * radius * 3.14159;
10
11     for (String s: args) {
12         radius = Float.parseFloat(s);
13         area = radius * radius * 3.14159;
14         System.out.println("Area of circle of radius " +
15                             radius + " = " + area);
16     }
17
18 }
19 }

```

ScannerExample.java

```

1 import javax.swing.JOptionPane;
2 import java.util.Scanner;
3
4 public class ScannerExample{
5     public static void main(String[] args){
6
7         Scanner input = new Scanner(System.in);
8
9         System.out.print("Enter an integer for seconds: ");
10        int seconds = input.nextInt();
11        int minutes = seconds/60;
12        int remainingSeconds = seconds % 60;
13
14        JOptionPane.showMessageDialog(null, seconds + " seconds
15                                     is " + minutes + " minutes and " + remainingSeconds
16                                     + " seconds", "ScannerExample", -1);
17    }
18 }

```

Lab 1

Lottery.java

```

1 package lottery;
2
3 import java.util.Scanner;
4
5 /**
6  *

```

```

7  * @author rv285
8  */
9  public class Lottery {
10
11      /**
12       * @param args the command line arguments
13       */
14      public static void main(String[] args) {
15
16          int randNum1, randNum2, userNumber;
17          randNum1 = (int) (Math.random() * 10);
18          randNum2 = (int) (Math.random() * 10);
19
20          Scanner s = new Scanner(System.in);
21
22          userNumber = s.nextInt();
23
24          int matches = 0;
25          while(userNumber != 0){
26
27              int digit = userNumber % 10;
28
29              if(digit == randNum1) matches++;
30              if(digit == randNum2) matches++;
31
32              userNumber /= 10;
33          }
34
35          if(matches == 2) System.out.println("You have won 3000
36              USD!");
37          else if(matches == 1) System.out.println("You have won
38              1000 USD!");
39          else System.out.println("Better Luck Next Time!");
40      }
41  }

```

Lottery2.java

```

1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4   */
5  package lottery;
6
7  import java.util.Scanner;

```

```

8 import javax.swing.*;
9
10
11 /**
12  *
13  * @author rv285
14  */
15 public class Lottery2 {
16
17     /**
18      * @param args the command line arguments
19      */
20     public static void main(String[] args) {
21         // TODO code application logic here
22         int randNum1, randNum2, userNumber;
23         randNum1 = (int) (Math.random() * 10);
24         randNum2 = (int) (Math.random() * 10);
25
26         //Scanner s = new Scanner(System.in);
27         //userNumber = s.nextInt();
28
29         //debug
30         //System.out.println(randNum1+(randNum2*10));
31
32         userNumber =
33             Integer.parseInt(JOptionPane.showInputDialog("Enter
34                 a two digit number"));
35
36         boolean exactMatch = false;
37
38         //check exact match
39         if(randNum1+(randNum2*10) == userNumber) exactMatch =
40             true;
41
42         if(exactMatch){
43             JOptionPane.showMessageDialog(null, "You have won
44                 10000 USD!");
45         }
46         //else
47         else{
48             int matches = 0;
49             while(userNumber != 0){
50
51                 int digit = userNumber % 10;
52
53                 if(digit == randNum1) matches+=1;

```

```

50         if(digit == randNum2) matches+=1;
51
52         userNumber /= 10;
53     }
54
55     if(matches >= 2) JOptionPane.showMessageDialog(null,
56         "You have won 3000 USD!");
57     else if(matches == 1)
58         JOptionPane.showMessageDialog(null, "You have
59         won 1000 USD!");
60     else JOptionPane.showMessageDialog(null, "Better
        Luck Next Time!");
    }
}
}

```

InsideCircle.java

```

1  /*
2  * To change this template, choose Tools / Templates
3  * and open the template in the editor.
4  */
5  package insidecircle;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author rv285
12   */
13  public class InsideCircle {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20
21          double x,y,r;
22          Scanner s = new Scanner(System.in);
23
24          x = s.nextDouble();
25          y = s.nextDouble();
26          r = s.nextDouble();
27
28          double LHS = Math.pow(x,2) + Math.pow(y,2);

```

```

29     double RHS = Math.pow(r,2);
30
31     if(LHS <= RHS){
32         System.out.println("The point is inside or on the
33                             circle");
34     }
35     else{
36         System.out.println("The point is outside the
37                             circle");
38     }
39 }

```

Quadratic.java

```

1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4   */
5  package quadratic;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author rv285
12   */
13  public class Quadratic {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          double a,b,c,d,r;
21
22          //a = 1;
23          Scanner s = new Scanner(System.in);
24
25          a = s.nextDouble();
26          b = s.nextDouble();
27          c = s.nextDouble();
28
29          d = Math.pow(b,2) - 4*a*c;
30

```

```

31     if(d == 0){
32         System.out.println("Root:");
33         r = (-b)/(2*a);
34         System.out.println(r);
35     }
36     else if (d > 0){
37         System.out.println("Roots:");
38
39         r = ((-b) + d)/(2*a);
40         System.out.println(r);
41
42         r = ((-b) - d)/(2*a);
43         System.out.println(r);
44     }
45 }
46 else {
47     System.out.println("The equation has no real
48         roots.");
49 }
50
51 }
52 }

```

Lab 2

Create a package SNU.geometryUtil. Define Classes For various 2-d shapes like Circle, Triangle, Square, Rectangle. Create variables, required constructors, methods for calculating areas, perimeters, for each type of shape. Create Mainclass having “ public static void main()” method in another package. Create menu driven program asking user to create a new object of any of the shape, show him the count of each object created, asking him to show any property of any of the object, calculate area or perimeter of any object.

Compiling

```
javac *.java -d .
```

Testing

```
java Test.Mainclass
```

Circle.java

```
rohan@rohan-K535V: ~/projects/monsoon/java/course/lab/3
New Object or Browse? <1,2,[0-exit]>
1
Choose Object type (Circle, Rectangle, Square, Triangle <0,1,2,3>
0
1.0
New Object or Browse? <1,2,[0-exit]>
1
Choose Object type (Circle, Rectangle, Square, Triangle <0,1,2,3>
0
2.0
New Object or Browse? <1,2,[0-exit]>
1
Choose Object type (Circle, Rectangle, Square, Triangle <0,1,2,3>
0
3.0
New Object or Browse? <1,2,[0-exit]>
2
Choose Object type (Circle, Rectangle, Square, Triangle <0,1,2,3>
0
Circle #0 -> Area: 3.1415 Perim: 6.283
Circle #1 -> Area: 12.566 Perim: 12.566
Circle #2 -> Area: 28.2735 Perim: 18.849
New Object or Browse? <1,2,[0-exit]>
```

Figure 1: Circle Example

```
1 package SNU.geometryUtil;
2
3 public class Circle{
4     private double radius;
5
6     private static int counter = 0;
7
8     public Circle(){
9         counter++;
10    }
11
12    public Circle(double x){
13        radius = x;
14        counter++;
15    }
16
17    public static int getCounter(){
18        return counter;
19    }
20
21    public double getRadius(){
22        return radius;
```



```

23     }
24
25     public void setRadius(double x){
26         radius = x;
27     }
28
29     public double getArea(){
30         return 3.1415 * radius * radius;
31     }
32     public double getPerimeter(){
33         return 2* 3.1415 * radius ;
34     }
35 }

```

Rectangle.java

```

1 package SNU.geometryUtil;
2
3
4 public class Rectangle{
5     private double l,b;
6
7     private static int counter = 0;
8
9     public Rectangle(){
10         counter++;
11     }
12
13     public Rectangle(double x, double y){
14         l = x;
15         b = y;
16         counter++;
17     }
18
19     public static int getCounter(){
20         return counter;
21     }
22
23     public double getL(){
24         return l;
25     }
26     public double getB(){
27         return b;
28     }
29
30     public void setL(double x){

```

```

31         l = x;
32     }
33     public void setB(double x){
34         b = x;
35     }
36
37     public double getPerimeter(){
38         return 2.0*(l+b);
39     }
40
41     public double getArea(){
42         return l*b;
43     }
44 }
45
46 }

```

Square.java

```

1 package SNU.geometryUtil;
2
3
4 public class Square{
5     private double a;
6
7     private static int counter = 0;
8
9     public Square(){
10         counter++;
11     }
12
13     public Square(double x){
14         a = x;
15         counter++;
16     }
17
18     public static int getCounter(){
19         return counter;
20     }
21
22     public double getA(){
23         return a;
24     }
25
26     public void setA(double x){
27         a = x;

```

```

28     }
29
30
31     public double getPerimeter(){
32         return 4.0*a;
33     }
34
35     public double getArea(){
36         return a*a;
37     }
38
39 }

```

Triangle.java

```

1 package SNU.geometryUtil;
2
3
4 public class Triangle{
5     private double a,b,c;
6
7     private static int counter = 0;
8
9     public Triangle(){
10         counter++;
11     }
12
13     public static int getCounter(){
14         return counter;
15     }
16
17     public Triangle(double x,double y,double z){
18         a = x;
19         b = y;
20         c = z;
21         counter++;
22     }
23
24     public double getA(){
25         return a;
26     }
27
28     public double getB(){
29         return b;
30     }
31

```

```

32     public double getC(){
33         return c;
34     }
35
36     public void setA(double x){
37         a = x;
38     }
39
40     public void setB(double x){
41         b = x;
42     }
43
44     public void setC(double x){
45         c = x;
46     }
47
48     public double getPerimeter(){
49         return a+b+c;
50     }
51
52     public double getArea(){
53         double p = getPerimeter()/2.0;
54
55         return Math.sqrt(p*(p-a)*(p-b)*(p-c));
56     }
57
58 }

```

Mainclass.java

```

1  package Test;
2
3  import SNU.geometryUtil.*;
4
5  import java.util.*;
6
7  public class Mainclass{
8
9      public static void main(String args[]){
10
11          Circle[] circles = new Circle[11]; //0
12
13          Rectangle[] rect = new Rectangle[11]; // 1
14
15          Square[] square = new Square[11]; //2
16

```

```

17     Triangle[] triangles = new Triangle[11]; //3
18
19
20     System.out.println("Welcome to Test Module for
        SNU.geometryUtil");
21
22     Scanner in = new Scanner(System.in);
23     int choice = -1;
24     while(choice != 0){
25         System.out.println("New Object or Browse?
            <1,2,[0-exit]>");
26         choice = in.nextInt();
27
28         if(choice == 0) System.exit(0);
29
30         System.out.println("Choose Object type (Circle,
            Rectangle, Square, Triangle <0,1,2,3>");
31         int object = in.nextInt();
32
33         if(object == 0) {
34             if(choice == 1){
35                 Double input = in.nextDouble();
36                 circles[Circle.getCounter()] = new
                    Circle(input);
37             }
38             if(choice == 2){
39                 for(int i = 0; i < Circle.getCounter(); i++){
40                     System.out.println("Circle #"+i+" -> "+
                        "Area: " + circles[i].getArea() + "
                        Perim: " +
                        circles[i].getPerimeter());
41                 }
42             }
43         }
44         else if(object == 1){
45             if(choice == 1){
46                 Double l = in.nextDouble();
47                 Double b = in.nextDouble();
48                 rect[Rectangle.getCounter()] = new
                    Rectangle(l,b);
49             }
50             if(choice == 2){
51                 for(int i = 0; i < Circle.getCounter(); i++){
52                     System.out.println("Rect #"+i+" -> "+
                        "Area: " + rect[i].getArea() + "
                        Perim: " + rect[i].getPerimeter());

```

```

53     }
54 }
55 }
56 else if(object == 2){
57     if(choice == 1){
58         Double a = in.nextDouble();
59
60         square[Square.getCounter()] = new Square(a);
61     }
62     if(choice == 2){
63         for(int i = 0; i < Square.getCounter(); i++){
64             System.out.println("Sqr #" + i + " -> " +
65                 "Area: " + square[i].getArea() + "
66                 Perim: " + square[i].getPerimeter());
67         }
68     }
69 }
70 else if(object == 3){
71     if(choice == 1){
72         Double a = in.nextDouble();
73         Double b = in.nextDouble();
74         Double c = in.nextDouble();
75
76         triangles[Triangle.getCounter()] = new
77             Triangle(a,b,c);
78     }
79     if(choice == 2){
80         for(int i = 0; i < Square.getCounter(); i++){
81             System.out.println("Sqr #" + i + " -> " +
82                 "Area: " + square[i].getArea() + "
83                 Perim: " + square[i].getPerimeter());
84         }
85     }
86 }
87 }
88 }
89 }
90 }

```

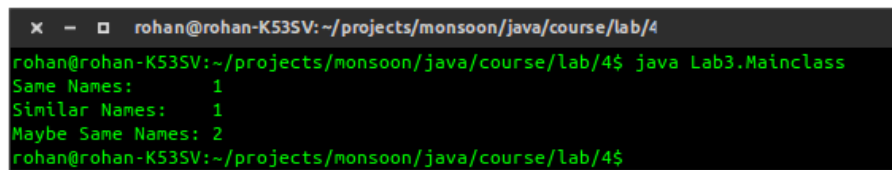
Lab 3

Define two classes.

1. **Name.** Having at least three components the First name, Middle Name and the Last Name. Add appropriate constructors.

Add methods to check sameNames, e.g. name1= Saurabh Kulshreshtha, ame2=Saurabh Kulshreshtha/ saurabh kulshreshtha.... sameInitials, e.g. name1=Ram Vashishtha, name2=Ravi Verma similarNames (When some of the first, middle or last names have been exchanged.),e.g. (Ravi Verma and Verma Ravi) or (Venkat Sai Velluri and Velluri Sai Venkat and Sai Velluri Venkat) etc. maybeSameNames (when some names contains initials in place of First, Middle and/or Last names and they are same as initials of other complete names) e.g. L. R. Ramesh, Luxmi Rama Ramesh,

2. **Student.** Has three names studentName, FathersName, MothersName. Constructors, Accessors/Mutators.
3. **Main Program.** Array of Students. Answer.... How many have same names, same fathers name etc..



```

rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/4
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/4$ java Lab3.Mainclass
Same Names: 1
Similar Names: 1
Maybe Same Names: 2
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/4$

```

Figure 2: Output of Lab3

Name.java

```

1 package Lab3;
2
3 import java.util.*;
4
5 public class Name{
6     String firstName;
7     String lastName;
8     String middleName;
9
10    public Name(String x){
11        String a[] = new String[3];
12        a = x.split(" ", 3);
13
14        if(a.length == 3){
15            firstName = a[0];
16            lastName = a[2];
17            middleName = a[1];
18        }

```

```

19     else if(a.length == 2){
20         firstName = a[0];
21         lastName = a[1];
22     }
23     else{
24         firstName = x;
25     }
26 }
27
28
29 public void setName(String x){
30     String a[] = new String[3];
31     a = x.split(" ", 3);
32
33     if(a.length == 3){
34         firstName = a[0];
35         lastName = a[2];
36         middleName = a[1];
37     }
38     else if(a.length == 2){
39         firstName = a[0];
40         lastName = a[1];
41     }
42     else{
43         firstName = x;
44     }
45 }
46
47 public String getName(){
48     String s = firstName;
49     if(middleName != null) s = s.concat(" " + middleName);
50     if(lastName != null) s = s.concat(" " + lastName);
51
52     return s;
53 }
54
55 public boolean sameName(Name x){
56     if(this.getName().toLowerCase().compareTo(x.getName().toLowerCase())
57         == 0) return true;
58     else return false;
59 }
60
61 public boolean similarName(Name x){
62     String[] a = new String[3];
63     a = x.getName().split(" ", 3);

```



```

64     Arrays.sort(a);
65
66     String[] b = new String[3];
67     b = this.getName().split(" ", 3);
68     Arrays.sort(b);
69
70     return Arrays.equals(a,b);
71 }
72
73 public boolean maybeSameName(Name x){
74
75     String[] a = new String[3];
76     a = x.getName().split(" ", 3);
77
78
79     //Arrays.sort(a);
80
81     String[] b = new String[3];
82     b = this.getName().split(" ", 3);
83     //    System.out.println(b[0]);
84
85     for(int i = 0; i < a.length; i++){
86         a[i] = a[i].substring(0,1);
87         //System.out.println(a[i]);
88     }
89     //Arrays.sort(b);
90
91     for(int i = 0; i < b.length; i++){
92         b[i] = b[i].substring(0,1);
93         //System.out.println(b[i]);
94     }
95     return Arrays.equals(a,b);
96 }
97
98 }

```

Student.java

```

1 package Lab3;
2
3 import java.util.*;
4
5 public class Student{
6     Name studentName = new Name("");
7     Name fathersName = new Name("");
8     Name mothersName = new Name("");

```

```

9
10     public Student(String s, String f, String m){
11         studentName.setName(s);
12         fathersName.setName(f);
13         mothersName.setName(m);
14     }
15
16     public Name getStudentName(){
17         return studentName;
18     }
19
20     public Name getFathersName(){
21         return fathersName;
22     }
23
24     public Name getMothersName(){
25         return mothersName;
26     }
27
28     public void setStudentName(String x){
29         studentName.setName(x);
30     }
31
32
33     public void setFathersName(String x){
34         fathersName.setName(x);
35     }
36
37     public void setMothersName(String x){
38         mothersName.setName(x);
39     }
40
41
42 }

```

Mainclass.java

```

1 package Lab3;
2
3 import java.util.*;
4
5 public class Mainclass{
6
7     public static void main(String args[]){
8
9         Student arr[] = new Student[5];

```

```

10
11     arr[0] = new Student("John Doe", "Bob Doe", "Lisa Doe");
12     arr[1] = new Student("john doe", "Bob Doe", "Lisa Doe");
13     arr[2] = new Student("joseph donalad", "Bob Doe", "Lisa
14         Doe");
15     arr[3] = new Student("Jack Daniels", "Bob Doe", "Lisa
16         Doe");
17     arr[4] = new Student("Doe John", "Bob Doe", "Lisa Doe");
18
19     int countSameName = 0, countSimilarName = 0,
20         countMaybeSameName = 0;
21
22     for(int i = 0; i < 5; i++){
23         for(int j = 0; j < 5; j++){
24             if(i != j && i < j){
25                 if(arr[i].getStudentName().sameName(arr[j].getStudentName()))
26                     countSameName++;
27                 if(arr[i].getStudentName().similarName(arr[j].getStudentName()))
28                     countSimilarName++;
29                 if(arr[i].getStudentName().maybeSameName(arr[j].getStudentName()))
30                     countMaybeSameName++;
31             }
32         }
33     }
34
35     System.out.println("Same Names:      " + countSameName);
36     System.out.println("Similar Names:   " +
37         countSimilarName);
38     System.out.println("Maybe Same Names: " +
39         countMaybeSameName);
40 }

```

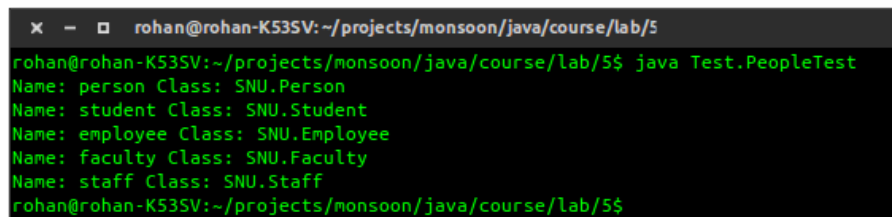
Lab 4

Questions

1. Design a class named Person and its two subclasses named Student and Employee . Make Faculty and Staff subclasses of Employee . A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Define a class named MyDate that contains the fields year , month , and day . A faculty member has office hours and a rank. A staff member has a title.

- Override the toString method in each class to display the class name and the person's name. Write a test program that creates a Person, Student, Employee, Faculty, and Staff, and invokes their toString() methods.
2. Define the Triangle class with three sides. In a triangle, the sum of any two sides is greater than the other side. The Triangle class must adhere to this rule. Create the IllegalTriangleException class, and modify the constructor of the Triangle class to throw an IllegalTriangleException object if a triangle is created with sides that violate the rule.

Source Files in Ascending order of filename.



```
rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/5
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/5$ java Test.PeopleTest
Name: person Class: SNU.Person
Name: student Class: SNU.Student
Name: employee Class: SNU.Employee
Name: faculty Class: SNU.Faculty
Name: staff Class: SNU.Staff
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/5$
```

Figure 3: Output of Part 1 (Person Class and it's children)

Employee.java

```
1 package SNU;
2
3 public class Employee extends Person {
4
5     protected double salary;
6     protected String office;
7     protected MyDate dateHired;
8
9     public Employee(String name) {
10         this(name, 0, "none", new MyDate());
11     }
12
13     public Employee(String name, double salary, String office,
14         MyDate dateHired) {
15         super(name);
16         this.salary = salary;
17         this.office = office;
18         this.dateHired = dateHired;
19     }
20     public double getSalary() {
```

```

21     return salary;
22 }
23
24 public void setSalary(double salary) {
25     this.salary = salary;
26 }
27
28 public String getOffice() {
29     return office;
30 }
31
32 public void setOffice(String office) {
33     this.office = office;
34 }
35
36 public MyDate getDateHired() {
37     return dateHired;
38 }
39
40 public void setDateHired(MyDate dateHired) {
41     this.dateHired = dateHired;
42 }
43
44 @Override
45 public String toString() {
46     return "Name: " + getName() + " Class: " +
47         this.getClass().getName();
48 }

```

Faculty.java

```

1 package SNU;
2
3 public class Faculty extends Employee {
4
5     public static String LECTURER = "Lecturer";
6     public static String ASSISTANT_PROFESSOR = "Assistant
7         Professor";
8     public static String ASSOCIATE_PROFESSOR = "Associate
9         Professor";
10    public static String PROFESSOR = "Professor";
11
12    protected String officeHours;
13    protected String rank;

```

```

13 public Faculty(String name) {
14     this(name, "9-5PM", "Employee");
15 }
16
17 public Faculty(String name, String officeHours, String rank)
18 {
19     super(name);
20     this.officeHours = officeHours;
21     this.rank = rank;
22 }
23
24 public String getOfficeHours() {
25     return officeHours;
26 }
27
28 public void setOfficeHours(String officeHours) {
29     this.officeHours = officeHours;
30 }
31
32 public String getRank() {
33     return rank;
34 }
35
36 public void setRank(String rank) {
37     this.rank = rank;
38 }
39
40 @Override
41 public String toString() {
42     return "Name: " + getName() + " Class: " +
43         this.getClass().getName();
44 }

```

GeometricObject.java

```

1 package SNU;
2
3 public abstract class GeometricObject implements
4     Comparable<GeometricObject> {
5     private String color = "white";
6     private boolean filled;
7     private java.util.Date dateCreated;
8
9     /** Construct a default geometric object */
10    protected GeometricObject() {

```

```

10     dateCreated = new java.util.Date();
11 }
12
13 /** Construct a geometric object with color and filled value
14 */
15 protected GeometricObject(String color, boolean filled) {
16     dateCreated = new java.util.Date();
17     this.color = color;
18     this.filled = filled;
19 }
20
21 /** Return color */
22 public String getColor() {
23     return color;
24 }
25
26 /** Set a new color */
27 public void setColor(String color) {
28     this.color = color;
29 }
30
31 /** Return filled. Since filled is boolean,
32 * the get method is named isFilled */
33 public boolean isFilled() {
34     return filled;
35 }
36
37 /** Set a new filled */
38 public void setFilled(boolean filled) {
39     this.filled = filled;
40 }
41
42 /** Get dateCreated */
43 public java.util.Date getDateCreated() {
44     return dateCreated;
45 }
46
47 /** Return a string representation of this object */
48 @Override
49 public String toString() {
50     return "created on " + dateCreated + "\n color: " + color
51         +
52         " and filled: " + filled;
53 }
54
55 @Override

```

```

54     public int compareTo(GeometricObject o) {
55         if (getArea() > o.getArea())
56             return 1;
57         else if (getArea() < o.getArea())
58             return -1;
59         else
60             return 0;
61     }
62
63     public static GeometricObject max(GeometricObject o1,
64         GeometricObject o2) {
65         return (o1.compareTo(o2) >= 0) ? o1 : o2;
66     }
67
68     public static double sumArea(GeometricObject[] a) {
69         double sum = 0;
70         for (GeometricObject o : a) {
71             sum += o.getArea();
72         }
73         return sum;
74     }
75     /** Abstract method getArea */
76     public abstract double getArea();
77
78     /** Abstract method getPerimeter */
79     public abstract double getPerimeter();
80 }

```

MyDate.java

```

1 package SNU;
2
3 import java.util.Calendar;
4 import java.util.GregorianCalendar;
5
6 public class MyDate {
7
8     private int year;
9     private int month;
10    private int day;
11
12    public MyDate() {
13
14        GregorianCalendar cal = new GregorianCalendar();
15        year = cal.get(Calendar.YEAR);
16        month = cal.get(Calendar.MONTH);

```



```

17         day = cal.get(Calendar.DAY_OF_MONTH);
18
19     }
20
21     public MyDate(long elapsedTime) {
22         GregorianCalendar cal = new GregorianCalendar();
23         cal.setTimeInMillis(elapsedTime);
24         year = cal.get(Calendar.YEAR);
25         month = cal.get(Calendar.MONTH);
26         day = cal.get(Calendar.DAY_OF_MONTH);
27     }
28
29     public MyDate(int year, int month, int day) {
30         this.year = year;
31         this.month = month;
32         this.day = day;
33     }
34
35     public int getYear() {
36         return year;
37     }
38
39     public void setYear(int year) {
40         this.year = year;
41     }
42
43     public int getMonth() {
44         return month;
45     }
46
47     public void setMonth(int month) {
48         this.month = month;
49     }
50
51     public int getDay() {
52         return day;
53     }
54
55     public void setDay(int day) {
56         this.day = day;
57     }
58
59     public void setDate(long elapsedTime) {
60         GregorianCalendar cal = new GregorianCalendar();
61         cal.setTimeInMillis(elapsedTime);
62         year = cal.get(Calendar.YEAR);

```

```

63     month = cal.get(Calendar.MONTH);
64     day = cal.get(Calendar.DAY_OF_MONTH);
65 }
66 }

```

PeopleTest.java

```

1 package Test;
2
3 import SNU.*;
4
5 public class PeopleTest {
6
7     public static void main(String[] args) {
8
9         Person person = new Person("person");
10        Student student = new Student("student");
11        Employee employee = new Employee("employee");
12        Faculty faculty = new Faculty("faculty");
13        Staff staff = new Staff("staff");
14
15        System.out.println(person.toString());
16        System.out.println(student.toString());
17        System.out.println(employee.toString());
18        System.out.println(faculty.toString());
19        System.out.println(staff.toString());
20
21    }
22
23
24 }

```

Person.java

```

1 package SNU;
2
3
4 public class Person {
5
6     protected String name;
7     protected String address;
8     protected String phoneNumber;
9     protected String email;
10
11     public Person(String name) {
12         this.name = name;

```

```

13     }
14
15     public String getName() {
16         return name;
17     }
18
19     public void setName(String name) {
20         this.name = name;
21     }
22
23     public String getAddress() {
24         return address;
25     }
26
27     public void setAddress(String address) {
28         this.address = address;
29     }
30
31     public String getPhoneNumber() {
32         return phoneNumber;
33     }
34
35     public void setPhoneNumber(String phoneNumber) {
36         this.phoneNumber = phoneNumber;
37     }
38
39     public String getEmail() {
40         return email;
41     }
42
43     public void setEmail(String email) {
44         this.email = email;
45     }
46
47     @Override
48     public String toString() {
49         return "Name: " + getName() + " Class: " +
50             this.getClass().getName();
51     }

```

Staff.java

```

1 package SNU;
2
3 public class Staff extends Employee {

```

```

4
5     protected String title;
6
7     public Staff(String name) {
8         this(name, "none");
9     }
10
11
12     public Staff(String name, String title) {
13         super(name);
14         this.title = title;
15     }
16
17     public String getTitle() {
18         return title;
19     }
20
21     public void setTitle(String title) {
22         this.title = title;
23     }
24
25     @Override
26     public String toString() {
27         return "Name: " + getName() + " Class: " +
28             this.getClass().getName();
29 }

```

Student.java

```

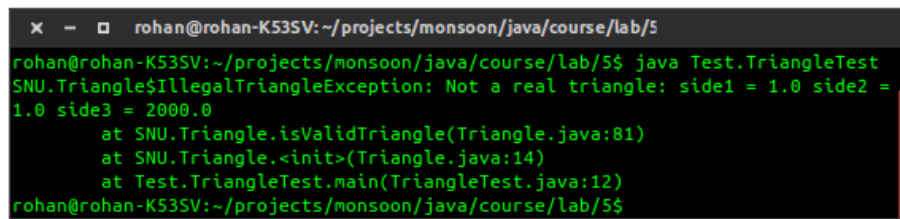
1 package SNU;
2
3 public class Student extends Person {
4
5     public static final String FRESHMAN = "Freshman";
6     public static final String SOPHOMORE = "Sophomore";
7     public static final String JUNIOR = "Junior";
8     public static final String SENIOR = "Senior";
9
10    protected String status;
11
12    public Student(String name) {
13        super(name);
14    }
15
16    public Student(String name, String status) {

```

```

17     super(name);
18     this.status = status;
19 }
20
21 @Override
22 public String toString() {
23     return "Name: " + getName() + " Class: " +
24         this.getClass().getName();
25 }

```



```

rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/5
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/5$ java Test.TriangleTest
SNU.Triangle$IllegalTriangleException: Not a real triangle: side1 = 1.0 side2 =
1.0 side3 = 2000.0
    at SNU.Triangle.isValidTriangle(Triangle.java:81)
    at SNU.Triangle.<init>(Triangle.java:14)
    at Test.TriangleTest.main(TriangleTest.java:12)
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/5$

```

Figure 4: Output of Part 2 (Triangle Class)

Triangle.java

```

1 package SNU;
2
3 public class Triangle extends GeometricObject {
4
5     private double side1;
6     private double side2;
7     private double side3;
8
9
10    public Triangle(double side1, double side2, double side3)
11        throws IllegalTriangleException {
12        this.side1 = side1;
13        this.side2 = side2;
14        this.side3 = side3;
15        isValidTriangle();
16    }
17
18    public Triangle() {
19        this.side1 = 1;
20        this.side2 = 1;
21        this.side3 = 1;

```

```

21     }
22
23     @Override
24     public double getArea() {
25
26         double s = (side1 + side2 + side3) / 2.0;
27         return Math.pow(s * (s - side1) * (s - side2) * (s -
28             side3), 0.5);
29     }
30
31     @Override
32     public double getPerimeter() {
33         return side1 + side2 + side3;
34     }
35
36     @Override
37     public String toString() {
38         return "Triangle{" +
39             "side1=" + side1 +
40             ", side2=" + side2 +
41             ", side3=" + side3 +
42             '}';
43     }
44
45     public static boolean isTriangle(double side1, double side2,
46         double side3) {
47
48         return ((side1 + side2 > side3) &&
49             (side1 + side3 > side2) &&
50             (side3 + side2 > side1));
51     }
52
53     public double getSide1() {
54         return side1;
55     }
56
57     public void setSide1(double side1) throws
58         IllegalTriangleException {
59         this.side1 = side1;
60         isValidTriangle();
61     }
62
63     public double getSide2() {
64         return side2;
65     }

```

```

64
65     public void setSide2(double side2) throws
        IllegalTriangleException{
66         this.side2 = side2;
67         isValidTriangle();
68     }
69
70     public double getSide3() {
71         return side3;
72     }
73
74     public void setSide3(double side3) throws
        IllegalTriangleException {
75         this.side3 = side3;
76         isValidTriangle();
77     }
78
79     private void isValidTriangle() throws
        IllegalTriangleException{
80         if (!isTriangle(side1, side2, side3)) {
81             throw new IllegalTriangleException(side1, side2,
                side3);
82         }
83     }
84
85     public class IllegalTriangleException extends
        IllegalArgumentException {
86
87         private double s1;
88         private double s2;
89         private double s3;
90
91         public IllegalTriangleException(double s1, double s2,
            double s3) {
92             super("Not a real triangle:" + " side1 = " + s1 + "
                side2 = " + s2 + " side3 = " + s3);
93             this.s1 = s1;
94             this.s2 = s2;
95             this.s3 = s3;
96         }
97
98         public double getS1() {
99             return s1;
100         }
101
102         public double getS2() {

```

```

103         return s2;
104     }
105
106     public double getS3() {
107         return s3;
108     }
109
110
111 }
112 }

```

TriangleTest.java

```

1 package Test;
2
3 import SNU.Triangle;
4 import SNU.Triangle.IllegalTriangleException;
5
6 public class TriangleTest {
7
8     public static void main(String[] args) {
9
10         Triangle t1;
11         try {
12             t1 = new Triangle(1, 1, 2000);
13         } catch (IllegalTriangleException e) {
14             e.printStackTrace();
15         }
16
17     }
18
19 }

```

Lab 5

1. Define the abstract GeometricObject as the base class for all classes like Circle, Rectangle, Triangle... defined earlier. It should have abstract method getArea and getPerimeter. Define a generic method costOfPaintingshape(GeometricObject g) and use to find cost for different geometric objects placed in an array based on the User.
2. (Enabling GeometricObject comparable) Modify the GeometricObject class to implement the Comparable interface, and define a static max method in the GeometricObject class for finding the larger of two GeometricObject objects. Draw the UML diagram and implement the new GeometricObject class. Write a test program that uses the max method to find the larger of two circles and the larger of two rectangles.

3. (The Colorable interface) Design an interface named Colorable with a void method named howToColor(). Every class of a colorable object must implement the Colorable interface. Design a class named Square that extends GeometricObject and implements Colorable. Implement howToColor to display a message “Color all four sides”. Draw a UML diagram that involves Colorable, Square, and GeometricObject.
4. Write a test program that creates an array of five GeometricObjects. For each object in the array, invoke its howToColor method if it is colorable.
5. (Finding the largest object) Write a method that returns the largest object in an array of objects. The method signature is: public static Object max(Comparable[] a) All the objects are instances of the Comparable interface. The order of the objects in the array is determined using the compareTo method. Write a test program that creates an array of ten strings, an array of ten integers, and an array of ten dates, and finds the largest string, integer, and date in the arrays.

```

rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/Strue/lab5
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/Strue/lab5$ javac -d . *.java
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/Strue/lab5$ java SNU.
SNU.Circle          SNU.GeometricObject  SNU.Rectangle
SNU.Colorable       SNU.GeoObj
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/Strue/lab5$ java SNU.GeoObj

Cost of painting is 392.69908169872417
Color the circumference
Color the circumference
Cost of painting is 60.0
Color all 4 sides
Color all 4 sides
obj2 is greater
obj1 is greater
obj2 is greater
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/Strue/lab5$

```

Figure 5: Output of Lab 5

Circle.java

```

1
2 package SNU;
3
4 public class Circle extends GeometricObject{
5     private double rad;
6
7     public Circle()
8     {

```

```

9      super();
10     }
11     public Circle(double rad, String color, boolean filled)
12     {
13         super(color,filled);
14         this.rad = rad;
15     }
16
17     public void setRadius(double rad)
18     {
19         this.rad = rad;
20     }
21     public double getRadius()
22     {
23         return rad;
24     }
25     public double getDiameter()
26     {
27         return rad*2.0;
28     }
29
30     /**
31      *
32      * @return
33      */
34     @Override
35     public double getArea()
36     {
37         return (rad*rad*Math.PI);
38     }
39
40     /**
41      *
42      * @return
43      */
44     @Override
45     public double getPerimeter()
46     {
47         return (2.0 * rad * Math.PI);
48     }
49     @Override
50     public void howToColor()
51     {
52         System.out.println("Color the circumference");
53     }
54

```

```
55 }
```

Colorable.java

```
1
2 package SNU;
3
4 public interface Colorable {
5
6     /**
7      *
8      */
9     public abstract void howToColor();
10
11 }
```

GeoObj.java

```
1 package SNU;
2
3 import static SNU.GeometricObject.max;
4 import java.util.Scanner;
5
6 public class GeoObj {
7
8     /**
9      * @param args the command line arguments
10     */
11     public static void main(String[] args) {
12         GeometricObject[] objects = {new Circle(5.0,"red",true),
13                                     new Circle(12.3,"",false), new
14                                     Rectangle(3.0,4.0,"blue",true), new
15                                     Rectangle(12.0,4.0,"",false)};
16         Scanner input = new Scanner(System.in);
17         int n,i;
18
19         for(i=0;i < objects.length;i++)
20         {
21             costOfPaintingAShape(objects[i]);
22             objects[i].howToColor();
23         }
24         for(i=0;i < objects.length - 1;i++)
25         {
26             max(objects[i],objects[i+1]);
27         }
28     }
29 }
```

```

26
27     // TODO code application logic here
28 }
29
30 public static void costOfPaintingAShape(GeometricObject obj)
31 {
32     double costPerUnitArea = 5.0, cost;
33
34     if(obj.isFilled())
35         System.out.println("Cost of painting is " +
36                             costPerUnitArea*obj.getArea());
37 }
38 }

```

GeometricObject.java

```

1
2 package SNU;
3
4 import java.util.Date;
5
6 public abstract class GeometricObject implements
7     Comparable<GeometricObject>,Colorable{
8     private String color;
9     private boolean filled;
10    private Date dateCreated;
11
12    protected GeometricObject()
13    {
14        dateCreated = new Date();
15    }
16    protected GeometricObject(String color, boolean filled)
17    {
18        dateCreated = new Date();
19        this.color = color;
20        this.filled = filled;
21    }
22
23    public String getColor()
24    {
25        return color;
26    }
27    public void setColor(String color)
28    {
29        this.color = color;
30    }
31 }

```

```

29     }
30     public boolean isFilled()
31     {
32         return filled;
33     }
34     public void setFilled(boolean filled)
35     {
36         this.filled = filled;
37     }
38     public Date getDate()
39     {
40         return dateCreated;
41     }
42     public static void max(GeometricObject obj1, GeometricObject
43         obj2)
44     {
45         System.out.println("obj"+obj1.compareTo(obj2)+" is
46             greater");
47     }
48     @Override
49     public int compareTo(GeometricObject o)
50     {
51         if(getArea() > o.getArea())
52             return 1;
53         else if(getArea() < o.getArea())
54             return 2;
55         else
56             return -1;
57     }
58     @Override
59     public abstract void howToColor();
60
61     public abstract double getArea();
62     public abstract double getPerimeter();
63 }

```

Rectangle.java

```

1
2 package SNU;
3
4 public class Rectangle extends GeometricObject implements
5     Colorable{
6     private double width,height;

```

```

6
7     public Rectangle()
8     {
9         super();
10    }
11    public Rectangle(double width, double height, String color,
12        boolean filled)
13    {
14        super(color,filled);
15        this.width = width;
16        this.height = height;
17    }
18
19    public void setWidth(double width)
20    {
21        this.width = width;
22    }
23    public double getWidth()
24    {
25        return width;
26    }
27    public void setHeight(double height)
28    {
29        this.height = height;
30    }
31    public double getHeight()
32    {
33        return height;
34    }
35
36    /**
37     *
38     * @return
39     */
40    @Override
41    public double getArea()
42    {
43        return (height*width);
44    }
45
46    /**
47     *
48     * @return
49     */
50    @Override
51    public double getPerimeter()

```

```

51     {
52         return (2.0 * (width+height));
53     }
54     @Override
55     public void howToColor()
56     {
57         System.out.println("Color all 4 sides");
58     }
59 }
60 }

```

Lab 6

Question 1

Write a program that draws a fixed circle, rectangle and triangle. Whenever a mouse is moved, display the message indicating whether the mouse point is inside the figure, as shown in figures below.

A.java (Circle)



Figure 6: Mouse Pointer is inside the Circle

```

1 package q1;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.MouseAdapter;
6 import java.awt.event.MouseEvent;

```



Figure 7: Mouse Pointer is outside the Circle

```

7
8 @SuppressWarnings("serial")
9 public class A extends JFrame {
10
11     private static JLabel jlabel = new JLabel("Mouse is
12         outside!");
13
14     private static int circleX=50, circleY=15, circleWidth=100,
15         centerX=circleX + circleWidth/2, centerY=circleY +
16         circleWidth/2;
17
18     public static double distance(
19         double x1, double y1, double x2, double y2) {
20         return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2
21             - y1));
22     }
23
24     public boolean isInsideCircle(int px, int py) {
25         return distance(px, py, centerX, centerY) <
26             circleWidth/2;
27     }
28
29     public A()
30     {
31         //Set JFrame title

```



```

29     super("Q1a");
30
31     //Set default close operation for JFrame
32     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33
34     //Set JFrame size
35     setSize(300,150);
36
37     //Make JFrame visible
38     setVisible(true);
39
40     this.addMouseMotionListener(new MouseAdapter() {
41
42         @Override
43         public void mouseMoved(MouseEvent e) {
44
45             super.mouseMoved(e);
46
47             //jlabel.setText(""+isInsideCircle(e.getX(),e.getY())+"
48                 "+e.getX()+" "+e.getY());
49
50             if(isInsideCircle(e.getX(),e.getY())){
51                 jlabel.setText("Mouse is inside!");
52             }
53             else{
54                 jlabel.setText("Mouse is outside!");
55             }
56
57             repaint();
58         }
59
60     });
61 }
62
63
64
65
66 public void paint(Graphics g)
67 {
68     super.paint(g);
69
70     g.drawOval(circleX, circleY, circleWidth, circleWidth);
71
72
73 }

```

```

74
75     public static void main(String[] args) {
76
77         A frame = new A();
78
79
80
81         frame.add(jlabel);
82
83     }
84
85 }

```

B.java (Rectangle)

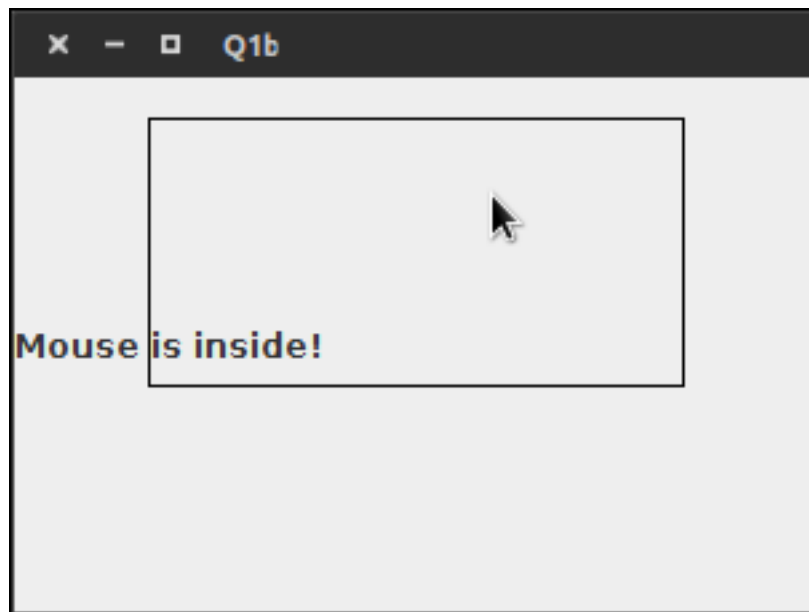


Figure 8: Mouse Pointer is inside the Rectangle

```

1 package q1;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.MouseAdapter;
6 import java.awt.event.MouseEvent;
7

```

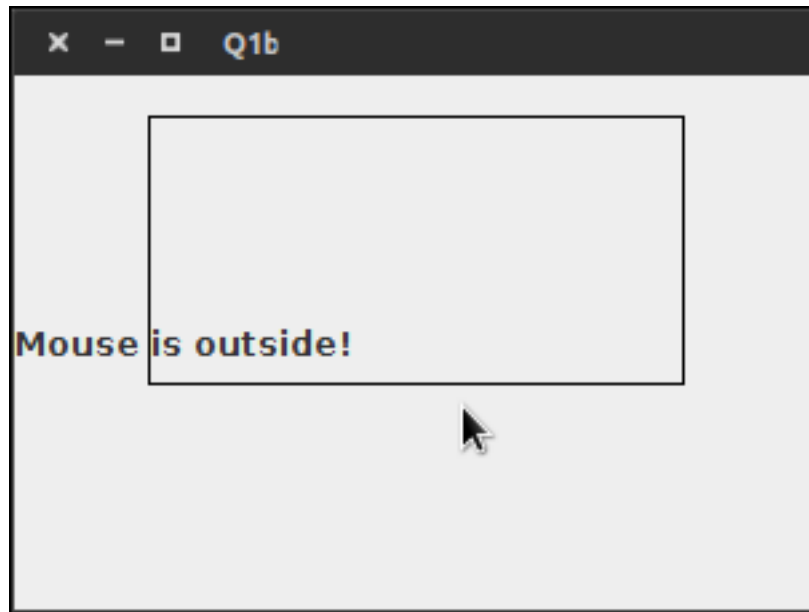


Figure 9: Mouse Pointer is outside the Rectangle

```
8 @SuppressWarnings("serial")
9 public class B extends JFrame {
10
11     private static JLabel jlabel = new JLabel("Mouse is
12         outside!");
13
14     private static int rectX=50, rectY=15, rectWidth=200,
15         rectHeight=100;
16
17     public static double distance(
18         double x1, double y1, double x2, double y2) {
19         return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2
20             - y1));
21     }
22
23     public boolean isInsideRect(int px, int py) {
24         return px < rectX+rectWidth && py < rectY+rectHeight &&
25             px > rectX && py > rectY;
26     }
27
28     public B()
29     {
```

```

27
28      //Set JFrame title
29      super("Q1b");
30
31      //Set default close operation for JFrame
32      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33
34      //Set JFrame size
35      setSize(300,200);
36
37      //Make JFrame visible
38      setVisible(true);
39
40      this.addMouseMotionListener(new MouseAdapter() {
41
42          @Override
43          public void mouseMoved(MouseEvent e) {
44
45              super.mouseMoved(e);
46
47              //jlabel.setText(""+isInsideRect(e.getX(),e.getY())+"
48                  "+e.getX()+" "+e.getY());
49
50              if(isInsideRect(e.getX(),e.getY())){
51                  jlabel.setText("Mouse is inside!");
52              }
53              else{
54                  jlabel.setText("Mouse is outside!");
55              }
56
57              repaint();
58          }
59
60      });
61  }
62
63
64
65
66  public void paint(Graphics g)
67  {
68      super.paint(g);
69
70      g.drawRect(rectX, rectY, rectWidth, rectHeight);
71

```

```

72     }
73
74     public static void main(String[] args) {
75
76         B frame = new B();
77
78
79
80         frame.add(jlabel);
81
82     }
83
84
85 }

```

C.java (Triangle)

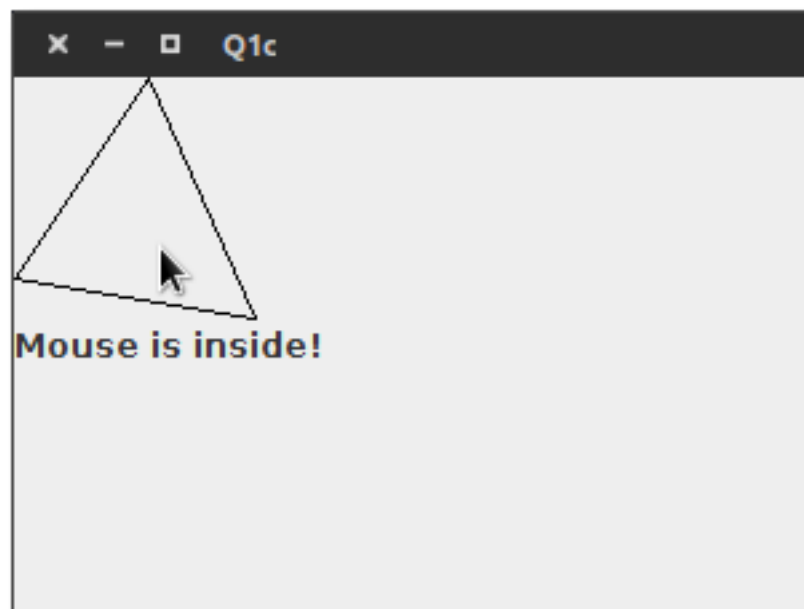


Figure 10: Mouse Pointer is inside the Tri

```

1 package q1;
2
3 import javax.swing.*;
4 import java.awt.*;

```

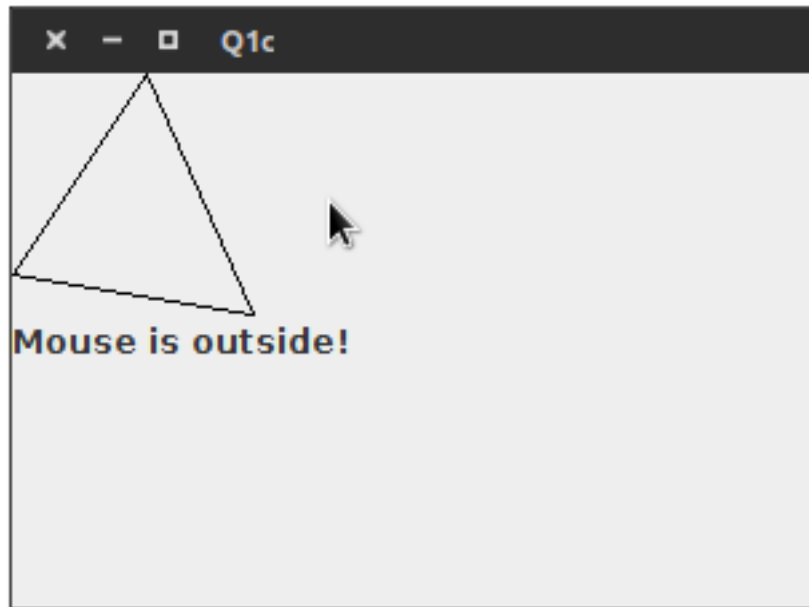


Figure 11: Mouse Pointer is outside the Tri

```

5 import java.awt.event.MouseAdapter;
6 import java.awt.event.MouseEvent;
7
8 import SNU.gr3.Point;
9
10 @SuppressWarnings("serial")
11 public class C extends JFrame {
12
13     private static JLabel jlabel = new JLabel("Mouse is
14         outside!");
15
16     private Point a = new Point(), b = new Point(), c = new Point();
17
18     private static double distance(
19         double x1, double y1, double x2, double y2) {
20         return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2
21             - y1));
22     }
23
24     private int sign(int n) {
25         return Math.abs(n) / n;
26     }
27 }

```

```

25
26     private int dotProduct(Point A, Point B, Point P){
27         return
28             (A.getX()-P.getX())*(B.getY()-P.getY())-(B.getX()-P.getX())*(A.getY()-P.getY());
29     }
30     public boolean isInside(int x, int y) {
31
32         Point p = new Point(x,y);
33
34         int ab = dotProduct(a,b,p);
35         int bc = dotProduct(b,c,p);
36         int ca = dotProduct(c,a,p);
37         return sign(ab)==sign(bc) && sign(bc)==sign(ca);
38
39     }
40
41     public C()
42     {
43
44
45         super("Q1c");
46
47         a.setX(0);
48         a.setY(75);
49         b.setX(90);
50         b.setY(90);
51         c.setX(50);
52         c.setY(0);
53
54         //Set JFrame title
55
56
57         //Set default close operation for JFrame
58         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
59
60         //Set JFrame size
61         setSize(300,200);
62
63         //Make JFrame visible
64         setVisible(true);
65
66         this.addMouseMotionListener(new MouseAdapter() {
67
68             @Override
69             public void mouseMoved(MouseEvent e) {

```

```

70
71         super.mouseMoved(e);
72
73         //jlabel.setText(""+isInside(e.getX(),e.getY())+"
74             "+e.getX()+" "+e.getY());
75
76         //repaint();
77
78         if(isInside(e.getX(),e.getY())){
79             jlabel.setText("Mouse is inside!");
80         }
81         else{
82             jlabel.setText("Mouse is outside!");
83         }
84
85         repaint();
86     }
87
88
89 });
90 }
91
92
93
94 public void paint(Graphics g)
95 {
96     super.paint(g);
97
98     g.drawLine(a.getX(),a.getY(),b.getX(),b.getY());
99     g.drawLine(b.getX(),b.getY(),c.getX(),c.getY());
100    g.drawLine(c.getX(),c.getY(),a.getX(),a.getY());
101
102 }
103
104 public static void main(String[] args) {
105
106     C frame = new C();
107
108     frame.add(jlabel);
109     frame.repaint();
110
111 }
112
113 }

```


Question 2

Write a program that displays a circle. You can point the mouse inside the circle and drag (i.e., move with mouse pressed) the circle wherever the mouse goes, as shown in Figure below.

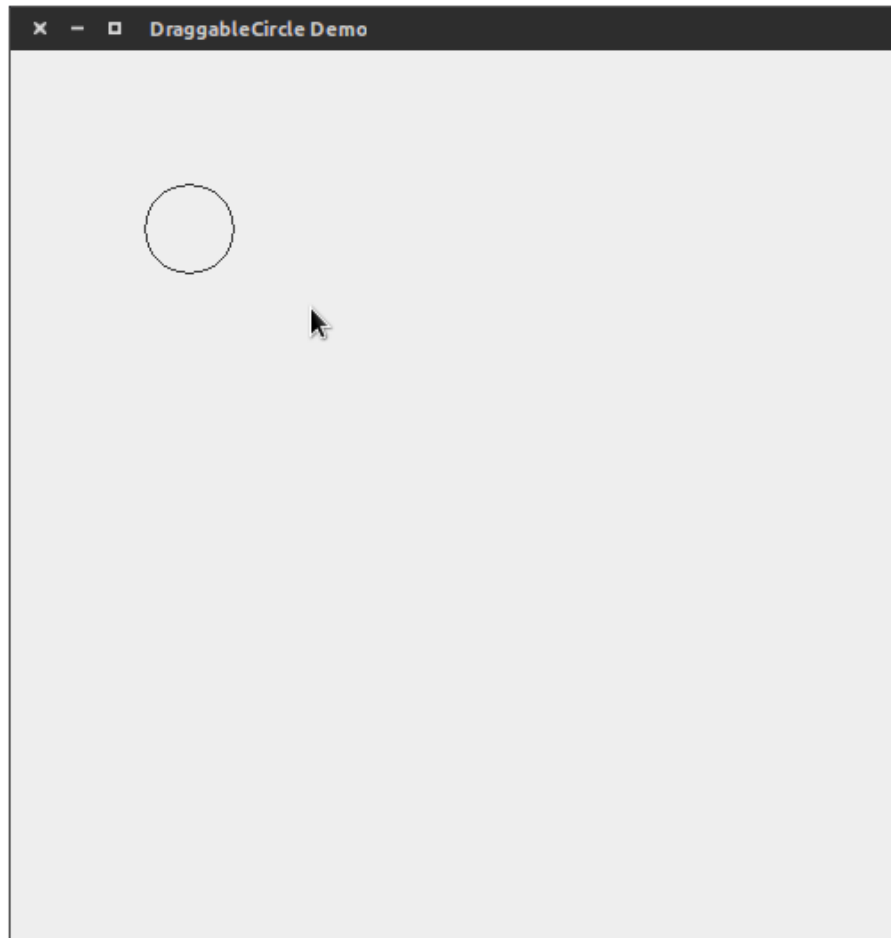


Figure 12: Initial State

DraggableCircle.java

```
1 package q2;  
2  
3 import java.awt.*;  
4 import java.awt.event.*;
```

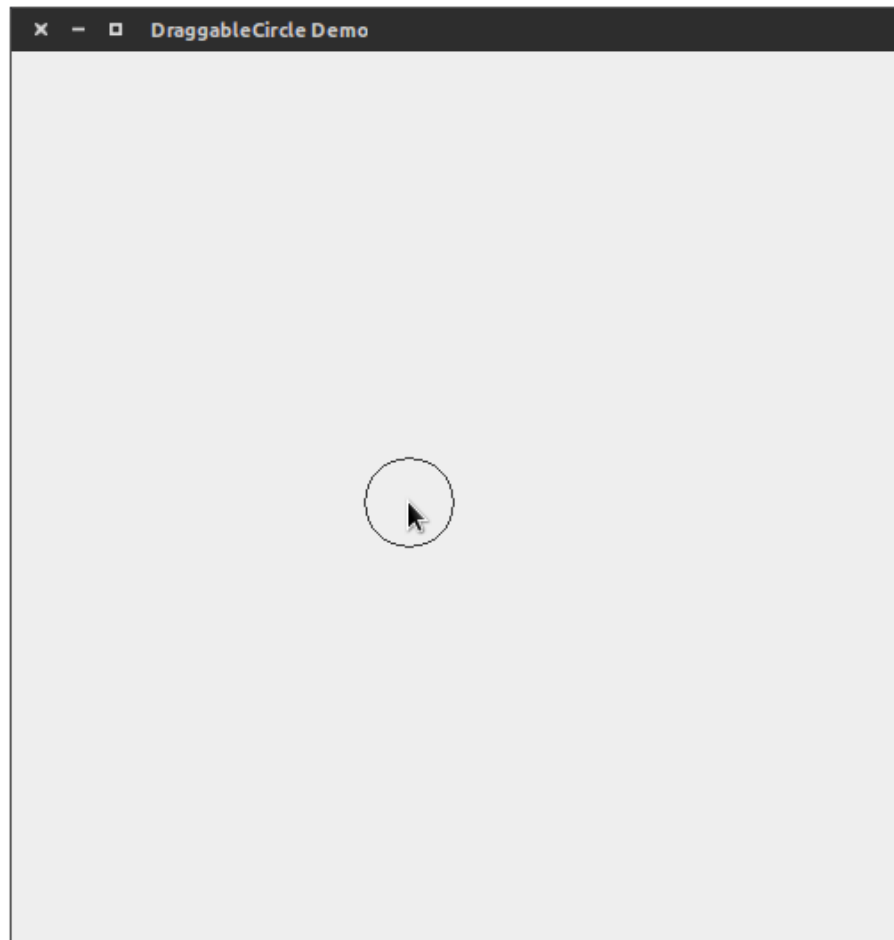


Figure 13: Dragging The Circle

```

5
6 import javax.swing.*;
7
8 @SuppressWarnings("serial")
9 public class DraggableCircle extends JFrame{
10
11     CirclePanel canvas = new CirclePanel();
12
13     public DraggableCircle() {
14         // TODO Auto-generated constructor stub
15
16         this.add(canvas);
17
18     }
19
20     class CirclePanel extends JPanel{
21         int x= 100, y = 100, diameter = 50;
22
23         public boolean isInsideCircle(int px, int py) {
24             return distance(px, py, x-diameter/2, y-diameter/2)
25                 < diameter;
26         }
27
28         public double distance(
29             double x1, double y1, double x2, double y2) {
30             return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) *
31                 (y2 - y1));
32         }
33
34         public CirclePanel() {
35             // TODO Auto-generated constructor stub
36             this.addMouseMotionListener(new MouseAdapter() {
37
38                 /*@Override
39                 public void mouseReleased(MouseEvent e) {
40                     // TODO Auto-generated method stub
41                     x = e.getX();
42                     y = e.getY();
43                     repaint();
44                 */
45                 /*
46                 @Override
47                 public void mousePressed(MouseEvent e) {
48                     // TODO Auto-generated method stub

```

```

49
50     */
51     @Override
52     public void mouseClicked(MouseEvent e) {
53         // TODO Auto-generated method stub
54         super.mouseClicked(e);
55         System.out.println("
56             "+isInsideCircle(e.getX(),e.getY()));
57     }
58     @Override
59     public void mouseDragged(MouseEvent e) {
60         // TODO Auto-generated method stub
61         super.mouseDragged(e);
62
63         if(isInsideCircle(e.getX(),e.getY()))
64         {
65             x = e.getX();
66             y = e.getY();
67             repaint();
68         }
69     }
70
71     });
72 }
73
74 @Override
75 protected void paintComponent(Graphics g) {
76     // TODO Auto-generated method stub
77     super.paintComponent(g);
78
79     g.drawOval(x-diameter/2, y-diameter/2, diameter,
80         diameter);
81 }
82
83 public static void main(String[] args) {
84     // TODO Auto-generated method stub
85     DraggableCircle frame = new DraggableCircle();
86     frame.setTitle("DraggableCircle Demo");
87     frame.setSize(500, 500);
88     frame.setVisible(true);
89 }
90
91 }

```

Question 3

Write a program that displays head (H) or tail (T) for each of nine coins, as shown in Figure below. When a cell is clicked, the coin is flipped. A cell is a `JLabel` . Write a custom cell class that extends `JLabel` with the mouse listener for handling the clicks. When the program starts, all cells initially display H .

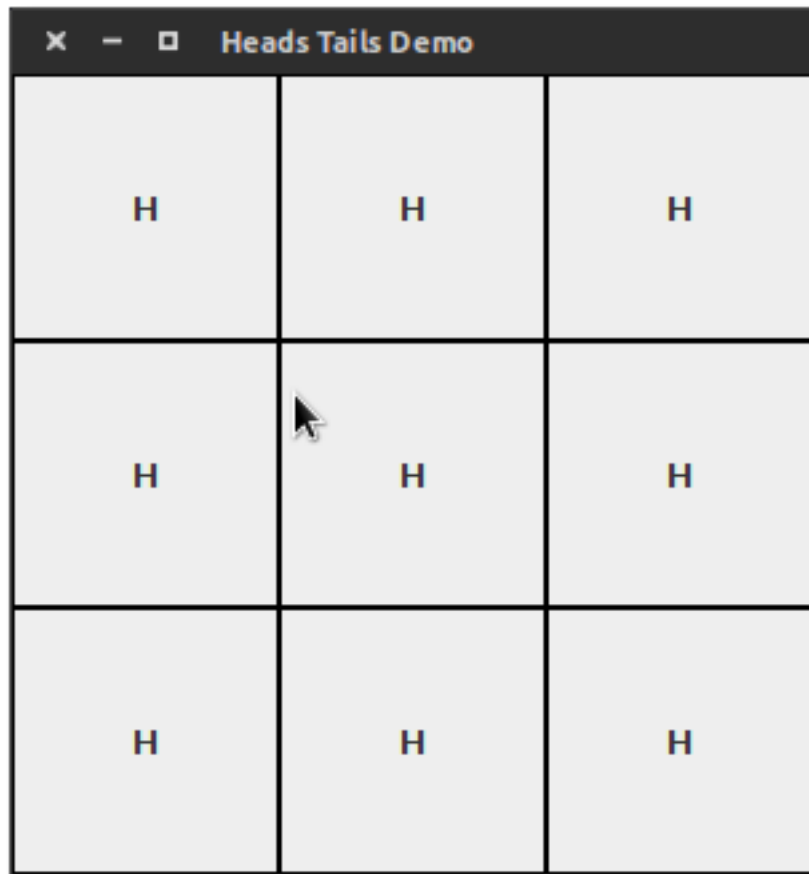


Figure 14: Initial Configuration

HeadsTails.java

```
1 package q3;  
2  
3 import java.awt.BorderLayout;  
4 import java.awt.Color;
```

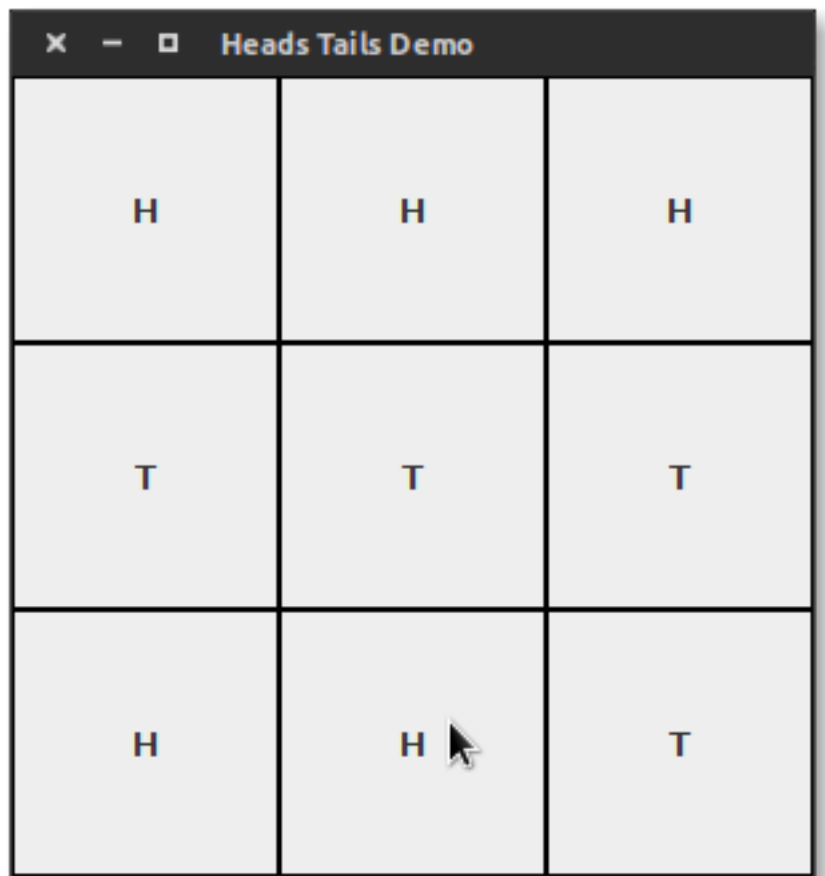


Figure 15: After Flipping

```

5 import java.awt.Font;
6 import java.awt.GridLayout;
7 import java.awt.event.MouseAdapter;
8 import java.awt.event.MouseEvent;
9
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14 import javax.swing.border.LineBorder;
15
16 @SuppressWarnings("serial")
17 public class HeadsTails extends JFrame{
18
19
20     public HeadsTails() {
21         // TODO Auto-generated constructor stub
22         setLayout(new GridLayout(3, 3));
23
24         for (int i = 0; i < 9; i++)
25             add(new ClickableCell("H"));
26     }
27
28     class ClickableCell extends JLabel {
29         public ClickableCell(String s) {
30             // TODO Auto-generated constructor stub
31             setBorder(new LineBorder(Color.black, 1));
32             setHorizontalAlignment(JLabel.CENTER);
33             setText(s);
34
35             addMouseListener(new MouseAdapter() {
36                 public void mouseClicked(MouseEvent e) {
37                     if (getText().equals("H")) {
38                         setText("T"); // Flip from H to T
39                     }
40                     else {
41                         setText("H"); // Flip from T to H
42                     }
43                 }
44             });
45         }
46     }
47
48
49     public static void main(String[] args) {
50         HeadsTails frame = new HeadsTails();

```

```

51
52     frame.setSize(300, 300);
53     frame.setTitle("Heads Tails Demo");
54     frame.setLocationRelativeTo(null);
55     frame.setVisible(true);
56
57 }
58
59 }

```

Lab 7

Question 1

Write a program to input a paragraph. Then can feed it in the program. Then perform the following using JCF: 1. find frequency of each vowel 2. Find and replace all occurrence of “the” with “THE”.

```

rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/7
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/7$ java ReplaceThe
Enter the String : This is the future of the world
This is THE future of THE world
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/7$ java VowelFrequency
Enter the String : This is the future of the world
Frequency of A : null
Frequency of E : 3
Frequency of I : 2
Frequency of O : 2
Frequency of U : 2
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/7$

```

Figure 16: Output of Question 1

VowelFrequency.java

```

1 import java.util.*;
2 class VowelFrequency
3 {
4     public static void main(String[] args)
5     {
6         Scanner in = new Scanner(System.in);
7         System.out.print("Enter the String : ");
8         String st = in.nextLine();
9         Map<String, Integer> map = new HashMap<String,
10             Integer>();

```



```

11     st=st.toLowerCase();
12
13     for(int i = 0;i<st.length();i++)
14     {
15         char c = st.charAt(i);
16
17         String ch = String.valueOf(c);
18
19         Integer frequency = map.get(ch);
20
21         Integer value;
22
23         if (frequency == null)
24             value = new Integer(1);
25         else{
26             value = new Integer(frequency.intValue()+1);
27         }
28         map.put(ch, value);
29
30
31     }
32
33     //System.out.println();
34     System.out.println("Frequency of A :
35         "+map.get(String.valueOf('a')));
36     System.out.println("Frequency of E :
37         "+map.get(String.valueOf('e')));
38     System.out.println("Frequency of I :
39         "+map.get(String.valueOf('i')));
40     System.out.println("Frequency of O :
41         "+map.get(String.valueOf('o')));
42     System.out.println("Frequency of U :
43         "+map.get(String.valueOf('u')));
44 }

```

ReplaceThe.java

```

1 import java.util.*;
2 class ReplaceThe
3 {
4     public static void main(String[] args)
5     {
6         Scanner in = new Scanner(System.in);
7         System.out.print("Enter the String : ");
8         String input = in.nextLine();

```

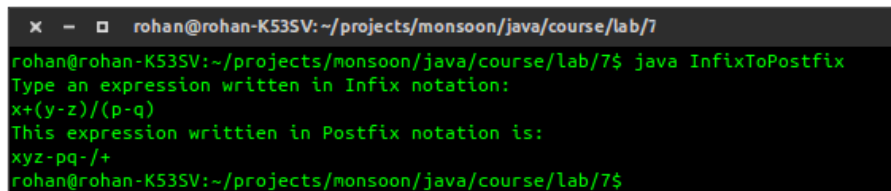
```

9
10     String[] st = input.split(" ");
11
12     List<String> list = new ArrayList<String>();
13
14     for(int i = 0; i < st.length;i++){
15         if(st[i].compareTo("the") == 0){
16             list.add(st[i].toUpperCase());
17         }
18         else{
19             list.add(st[i]);
20         }
21     }
22
23     String listString = "";
24
25     for (String s : list)
26     {
27         listString += s + " ";
28     }
29
30     System.out.println(listString);
31 }
32 }

```

Question 2

Write a program to evaluate an infix expression (input and output using JOption-pane) using JCF. Use only JCF interfaces and methods for all stack operations needed to solve the problem.



```

rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/7
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/7$ java InfixToPostfix
Type an expression written in Infix notation:
x+(y-z)/(p-q)
This expression wrtitten in Postfix notation is:
xyz-pq-/+
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/7$

```

Figure 17: Output of InfixToPostfix.java

```

1 import java.util.Scanner;
2 import java.util.Stack;
3
4 public class InfixToPostfix {
5     static private String expression;

```

```

6     private Stack<Character> stack = new Stack<Character>();
7
8     public InfixToPostfix(String infixExpression) {
9         expression = infixExpression;
10    }
11
12    public String infixToPostfix() {
13        String postfixString = "";
14
15        for (int index = 0; index < expression.length();
16             ++index) {
17            char value = expression.charAt(index);
18            if (value == '(') {
19                stack.push('(');
20            } else if (value == ')') {
21                Character oper = stack.peek();
22
23                while (!(oper.equals('(')) &&
24                       !(stack.isEmpty())) {
25                    stack.pop();
26                    postfixString += oper.charValue();
27                    if (!stack.isEmpty())
28                        oper = stack.peek();
29                }
30                stack.pop();
31            } else if (value == '+' || value == '-') {
32                if (stack.isEmpty()) {
33                    stack.push(value);
34                } else {
35                    Character oper = stack.peek();
36                    while (!(stack.isEmpty() ||
37                           oper.equals('(') ||
38                           oper.equals(')'))) {
39                        oper = stack.pop();
40                        postfixString += oper.charValue();
41                    }
42                    stack.push(value);
43                }
44            } else if (value == '*' || value == '/') {
45                if (stack.isEmpty()) {
46                    stack.push(value);
47                } else {
48                    Character oper = stack.peek();
49
50                    while (!(oper.equals('(') &&
51                           !oper.equals('(') &&

```

```

47         !oper.equals('(',-')) && !stack.isEmpty())
48         {
49             oper = stack.pop();
50             postfixString += oper.charValue();
51         }
52         stack.push(value);
53     }
54     } else {
55         postfixString += value;
56     }
57 }
58
59 while (!stack.isEmpty()) {
60     Character oper = stack.peek();
61     if (!oper.equals('(',-')) {
62         stack.pop();
63         postfixString += oper.charValue();
64     }
65 }
66
67 return postfixString;
68 }
69
70 public static void main(String[] args) {
71     System.out.println("Type an expression written in Infix
72     notation: ");
73     Scanner input = new Scanner(System.in);
74     String expression = input.next();
75     InfixToPostfix convert = new InfixToPostfix(expression);
76     System.out.println("This expression writtten in Postfix
77     notation is: \n" + convert.infixToPostfix());
78 }
79 }

```

Lab Extra

Bio.java

```

1 import java.io.PrintStream;
2 import java.util.Scanner;
3 public class Bio
4 {
5 {
6
7     public static void main(String[] args)

```

```
rohan@rohan-K53SV: ~/projects/monsoon/java/course/lab/extra
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/extra$ cat input.io
3
TTATGTTTTAAGGATGGGGCGTTAGTT
TTATGGGGTAAGGATGGGGCGTTAGTT
TGTGTGTATAT
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/extra$ java Bio < input.io
> out.io
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/extra$ cat out.io
Line 1
TTT
GGGCGT
Line 2
GGGCGT
Line 3
no gene is found
rohan@rohan-K53SV:~/projects/monsoon/java/course/lab/extra$
```

Figure 18: Output of Bio.java

```
8      {
9          Scanner input = new Scanner(System.in);
10
11          int to = input.nextInt();
12          input.nextLine();
13          int t = to;
14          while(t-- != 0)
15          {
16              System.out.println("Line " + (to - t));
17              String genome = input.nextLine();
18              boolean found = false;
19              int start = -1;
20              for (int i = 0; i < genome.length() - 2; i++) {
21                  String triplet = genome.substring(i, i + 3);
22                  if (triplet.equals("ATG"))
23                  {
24                      start = i + 3;
25                  }
26                  else if (((triplet.equals("TAG")) ||
27                          (triplet.equals("TAA")) ||
28                          (triplet.equals("TGA")))) && (start != -1))
29                  {
30                      String gene = genome.substring(start, i);
31                      if (gene.length() % 3 == 0)
32                      {
33                          found = true;
34                          System.out.println(gene);
35                          start = -1;
36                      }
37                  }
38              }
39              if (!found)
40              {
41                  System.out.println("no gene is found");
42              }
43          }
44      }
```

```
34         }
35     }
36 }
37 if (!found)
38     System.out.println("no gene is found");
39 }
40 }
41 }
```

Fin.