**Instructions:**

Java files will be compiled and tested using the following syntax on a Windows desktop:

*javac Problem1.java*

*java Problem1*

All java files must compile using the command line and using java version 1.8.0\_151

Each java file submitted is to include commented code as well as your name, course number and date.

All java source files, and other documents, should be submitted in a zip package. For Homework 3, you will submit one file called homework3.zip to d2l. The file homework3.zip should contain the following files:

Problem1.java

Problem1UML.doc

Problem2.java

Problem2UML.doc

Problem2.java

Problem3.java

Problem4.java

Problems will be graded on the following:

* Commenting of code including but not limited to variables, classes
* Functions correctly based on tested use cases
* Neatly formatted code and conciseness.

Programs that do not compile will not receive credit.

**Problem 1**

Design a class named MyInteger. The class contains:

* An int data field named value that stores the int value represented by this object.
* A constructor that creates a MyInteger object for the specified int value.
* A get method that returns the int value.
* Methods isEven(), isOdd(), and isPrime() that return true if the value is even, odd, or prime, respectively.
* Static methods isEven(int), isOdd(int), and isPrime(int) that return true if the specified value is even, odd, or prime, respectively.
* Static methods isEven(MyInteger), isOdd(MyInteger), and isPrime(MyInteger) that return true if the specified value is even, odd, or prime, respectively.
* Methods equals(int) and equals(MyInteger) that return true if the value in the object is equal to the specified value.
* A static method parseInt(char[]) that converts an array of numeric characters to an int value.
* A static method parseInt(String) that converts a string into an int value.

For this problem you will need a UML diagram for the class. Please include this in your submitted zip file in a document called Problem1UML.doc

Submit your completed source code in a file called Problem1.java

Here is some pseudo code to get you started:

public class Problem1 {

    public static void main(String[] args) {

MyInteger n1 = new MyInteger(Integer.parseInt(args[0]);

MyInteger n2 = new MyInteger(Integer.parseInt(args[1]);

        System.out.println("n1 is even? " + n1.isEven());

        System.out.println("n1 is prime? " + n1.isPrime());

        char[] chars = { '9', '8', '7', '6' };

        System.out.println(MyInteger.parseInt(chars));

        String s = "5000";

        System.out.println(MyInteger.parseInt(s));

        System.out.println("n2 is odd? " + n2.isOdd());

        System.out.println("n1 is equal to n2? " + n1.equals(n2));

        System.out.println("n1 is equal to 5? " + n1.equals(5));

    }

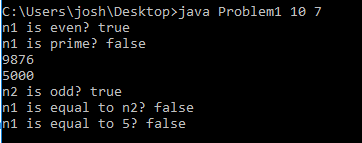
}

class MyInteger {

    // Implement your class here

}

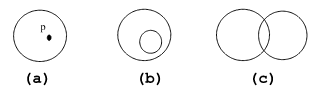
Here is a sample run of the program:



**Problem 2**

Define the Circle2D class that contains:

* 3 command line arguments, first for center x coordinate, second for center y coordinate and third for the radius of the circle.
* Two double data fields named x and y that specify the center of the circle with get methods.
* A data field radius with a get method.
* A no-arg constructor that creates a default circle with (0, 0) for (x, y) and 1 for radius.
* A constructor that creates a circle with the specified x, y, and radius.
* A method getArea() that returns the area of the circle.
* A method getPerimeter() that returns the perimeter of the circle.
* A method contains(double x, double y) that returns true if the specified point (x, y) is inside this circle.
* A method contains(Circle2D circle) that returns true if the specified circle is inside this circle. See Figure 10.14(b).
* A method overlaps(Circle2D circle) that returns true if the specified circle overlaps with this circle. See the figure below.



*(a) A point is inside the circle. (b) A circle is inside another circle. (c) A circle overlaps another circle.*

For this problem you will need:

1. A UML diagram for the class. Please include this in your submitted zip file in a document called Problem2UML.doc
2. A test program that creates a Circle2D object c1 (new Circle2D(2, 2, 5.5)), displays its area and perimeter, and displays the result of c1.contains(3, 3), c1.contains(new Circle2D(4, 5, 10.5)), and c1.overlaps(new Circle2D(3, 5, 2.3)). Other coordinates and radiuses will be tested. Make sure your program can handle values from 1 - 100. Please include this in your submitted zip file called Problem2.java

Here is some pseudo code to get you started:

*public class Problem2 {*

*public static void main(String[] args) {*

*if (args.length == 0) {*

*System.out.println("Please supply x y and radius arguments");*

*System.exit(0);*

*}*

*Double xval = Double.parseDouble(args[0]);*

*Double yval = Double.parseDouble(args[1]);*

*Double radius = Double.parseDouble(args[2]);*

*Circle2D c1 = new Circle2D(xval, yval, radius);*

*System.out.println("Area is " + c1.getArea());*

*System.out.println("Perimeter is " + c1.getPerimeter());*

*System.out.println(c1.contains(3, 3));*

*System.out.println(c1.contains(new Circle2D(4, 5, 10.5)));*

*System.out.println(c1.overlaps(new Circle2D(3, 5, 2.3)));*

*}*

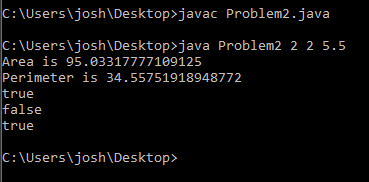
*}*

*class Circle2D {*

*//Implement your class here*

*}*

Here is a sample run of the program:



**Problem 3**

Design a class named Triangle that extends GeometricObject. You can download GeometricObject.java [here](https://drive.google.com/open?id=1vEU2a592nB0bEK2yOjM2lEgovI8PrecJ). The Triangle class contains:

* Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
* A no-arg constructor that creates a default triangle.
* A constructor that creates a triangle with the specified side1, side2, and side3.
* The accessor methods for all three data fields.
* A method named getArea() that returns the area of this triangle.
* A method named getPerimeter() that returns the perimeter of this triangle.
* A method named toString() that returns a string description for the triangle.

For the formula to compute the area of a triangle, see Exercise 5.19. The toString() method is implemented as follows:

return "Triangle: side1 = " + side1 + " side2 = " + side2 + " side3 = " + side3;

Here is some sample code to get you started:

import java.util.Scanner;

public class Problem3 {

 public static void main(String[] args) {

   Scanner input = new Scanner(System.in);

   System.out.print("Enter three sides: ");

   double side1 = input.nextDouble();

   double side2 = input.nextDouble();

   double side3 = input.nextDouble();

   Triangle triangle = new Triangle(side1, side2, side3);

   System.out.print("Enter the color: ");

   String color = input.next();

   triangle.setColor(color);

   System.out.print("Enter a boolean value for filled: ");

   boolean filled = input.nextBoolean();

   triangle.setFilled(filled);

   System.out.println("The color of the triangle is: " + triangle.getColor());

   System.out.println("The triangle is filled? " + triangle.isFilled());

   System.out.println("The area is " + triangle.getArea());

   System.out.println("The perimeter is "

     + triangle.getPerimeter());

   System.out.println(triangle);

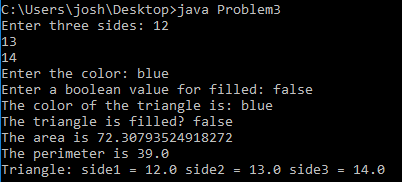
 }

}

class Triangle extends GeometricObject {

}

Here is a sample run of Program3



**Problem 4**

Files containing the popularity ranking of baby names from 2001 to 2010 are stored in files [here](https://drive.google.com/open?id=1nD4n5mESdWtaYJPE3yGcl8PFV9YYZkMM). Each line contains a ranking, a boy’s name, number for the boy’s name, a girl’s name, and number for the girl’s name. For example, the first two lines in the file babynameranking2010.txt are as follows:

1     Jacob    21,875     Isabella     22,731

2     Ethan    17,866     Sophia     20,477

So, the boy’s name Jacob and girl’s name Isabella are ranked #1 and the boy’s name Ethan and girl’s name Sophia are ranked #2. 21,875 boys are named Jacob and 22,731 girls are named Isabella. Write a program that prompts the user to enter the year, gender, and baby name then displays the ranking of the name for the year. Here are some sample runs:

