CSC171 — Homework 9

Naming & Scope

The goal of this assignment is to give you additional experience understanding the role of naming and scope in writing and understanding programs. You may want to refer to the scope rules shown in the textbook and discussed in class.

Questions

1. For each of the following programs, determine whether or not they are valid Java programs **without typing them in**. If they are valid, indicate what they output. If not, explain why (be precise).

   (a)
   ```
   public class Shadow1 {
       public static void main(String[] args) {
           int i = 0;
           System.out.println(i);
           int j = 1;
           System.out.println(j);
           int i = j;
           System.out.println(i);
       }
   }
   ```

   (b)
   ```
   public class Shadow2 {
       public static void main(String[] args) {
           int i;
           for (int i = 0; i < 10; i++) {
               System.out.println(i);
           }
       }
   }
   ```

   (c)
   ```
   public class Shadow3 {
       static private int i = 99;
       public static void main(String[] args) {
           for (int i = 0; i < 10; i++) {
               System.out.println(i);
           }
           System.out.println(i);
       }
   }
   ```

(d)

```
public class Shadow4 {
    private int i;
    public void doSomething(int n) {
        for (int i = 0; i < n; i++) {
            System.out.println(i);
        }
    }
    public static void main(String[] args) {
        new Shadow4().doSomething(9);
    }
}
```

(e)

```
public class Shadow5 {
    private int i;
    public void doSomething(int i) {
        for (int i = 0; i < 10; i++) {
            System.out.println(i);
        }
    }
    public static void main(String[] args) {
        new Shadow5().doSomething(9);
    }
}
```

(f)

```
public class Shadow6 {
    private int i;
    public void doSomething(int i) {
        for (i = 0; i < 10; i++) {
            System.out.println(i);
        }
    }
    public static void main(String[] args) {
        new Shadow6().doSomething(9);
    }
}
```

(g)

```
public class Shadow7 {
    private int i = 99;
    public void doSomething(int i) {
        for (i = 0; i < 10; i++) {
            System.out.println(i);
        }
    }
    public static void main(String[] args) {
        Shadow7 i = new Shadow7();
        i.doSomething(99);
        System.out.println(i.i);
    }
}
```

2. Without typing it in and running it, what is the output of the following program? Of course we can't tell if you type it in or not. But remember that on a quiz or exam, you won't be able to type it in. So the goal is to understand *why* it prints what it does. Test your answer by running the code. If you got the wrong output, use `println` statements to trace the program's execution until you understand it. You can also make up your own variations and test yourself further.

```
public class ScopeTester {

    protected int num;
    protected String str;
    static final String[] labels = { "Hello", "Goodbye", "Whatever" };

    public ScopeTester(int num, String str) {
        this.num = num;
        this.str = str;
    }

    public void printout() {
        System.out.println(num + " " + str);
    }

    public void doubleMe() {
        num *= 2;
        str += " " + str;
    }

    public int twoTimes(int num) {
        num *= 2;
        return num;
    }

    public static void main(String[] args) {
        for (int num=0; num < 3; num++) {
            ScopeTester tester = new ScopeTester(num, labels[num]);
            int num2 = tester.twoTimes(num);
            tester.doubleMe();
            tester.printout();
        }
    }

}
```

3. Develop three classes representing “players”: soccer players, saxophone players, and MP3 players. Your classes should be in a package entitled `my.players`. Write a `main` method in a separate class that creates and prints instances of the three classes of players. This “main” class should not be in any package (a.k.a. the unnamed or default package).

4. Use packages to implement two classes both named `Account`: one should represent bank accounts, the other should represent computer user accounts. Write a `main` method in another class (in the default package) that creates and prints an instance of each class. Hint: You can’t `import` two classes with the same name, but there is another way to refer to classes in other packages. Second hint: lookup “fully qualified name” in the Java Tutorial (or elsewhere).

5. Extra practice and review (not about naming and scope): Write a class `TriangleTable`. A `TriangleTable` is defined by an integer $n > 0$. There are n rows in the table. Row i (starting from 0) contains the numbers from i down to 0. For example, a `TriangleTable` of size 4 would look like the following:

   ```
   0
   1 0
   2 1 0
   3 2 1 0
   ```

   Your class should have a method that prints the instance nicely (like above). Your program should create and print tables of size from 1 to 10.

Grading Scheme

Equal weight for each part.

Doesn’t compile or is trivial	$<$ 50%
Compiles and is non-trivial	$\geq$ 50%
Complete and correct with good style and comments	100%
Incomplete, incorrect, bad style, no comments	$<$ 100%

Submission Requirements

Your submission **MUST** include a file named “`README.txt`” with your name, your NetID, the assignment number, and your lab section. This file should explain anything we need to know about how to build and run your project. In particular, be sure to explain how to run what parts of your submission for each question in the assignment.

Submit your solution as a single ZIP archive to BlackBoard before the deadline.

Late homeworks will not be graded and will receive a grade of 0.

All assignments and activities associated with this course must be performed in accordance with the University of Rochester’s Academic Honesty Policy.