Object-Oriented Programming in Java
IFT 194: HW 3

Brandon Doyle
bdoyle5@asu.edu
1215232174

Dr. Usha Jagannathan
Usha.Jagannathan@asu.edu

July 16, 2018

Summary

Homework 3.1

3.3

In this question we're asked to declare a `String` variable and initialize it to contain the same characters as another, except in all uppercase characters.

```
...
String name = "Brandon Doyle";
String str = name.toUpperCase();
...
```

3.8

We're tasked with using the `java.util.Random` class to generate random numbers in the specified ranges. See my solution in Figure 1.

```
package hw_3;

import java.util.Random;

public class Rand
{
    public static void main(String[] args)
    {
        var rand = new Random();

        // a.
        var randA = rand.nextInt(11);

        // b.
        var randB = rand.nextInt(401);

        // c.
        var randC = rand.nextInt(10) + 1;

        // d.
        var randD = rand.nextInt(400) + 1;

        // e.
        var randE = rand.nextInt(26) + 25;

        // f.
        var randF = rand.nextInt(26) - 10;

        System.out.println(randA + " " + randB + " " + randC + " " + randD + " "
                               + randE + " " + randF);
    }
}
```

Figure 1: Rand.java

View the source of this document on GitHub.

3.9

In this question we're asked to write an expression that computes the square root of a sum and stores the result in a variable. I would write this as follows.

```
...
double num1 = 1.0, num2 = 2.0;
double num3 = Math.sqrt(num1 + num2);
...
```

3.10

In this question we're asked to write a single statement that computes the absolute value of a variable `total`. I would write this as follows.

```
var absTotal = Math.abs(total);
```

Note also that I've used local variable type inference, because there exists an overloaded method for `long`, `int`, `float`, and `double` types.

Alternatively, we might write something as follows, which does not use any library methods.

```
var absTotal = (total < 0) ? -total : total;
```

3.11

In this problem we're asked to write code that will create a `DecimalFormaat` object to round a value to 4 decimal places. Please see my solution in Figure 2.

```
package hw_3;

import java.text.DecimalFormat;

public class Formatter
{
    public static void main(String[] args)
    {
        double pi = 3.14159265;
        var df = new DecimalFormat("#.####");
        String result = df.format(pi);
        System.out.println(result);
    }
}
```

Figure 2: Formatter.java

This program outputs `3.1416` to my console.

Something I came across that I found rather interesting is there's no need to call `String.format` in `System.out.println` because we also have `System.out.printf`, which is very similar to C's `printf` function (defined in `stdio.h`). I'll be sure to remember this while writing future labs and homework.

3.12

In this problem we're tasked with obtaining a double from the user and printing that value to the fourth power to 3 decimal places. Please see Figure 3 for my solution.

```java
package hw_3;

import java.util.Scanner;
import java.util.InputMismatchException;

public class GetOutput
{
    public static void main(String[] args)
    {
        try (var scnr = new Scanner(System.in)) {
            double value = getDouble(scnr);

            double power = Math.pow(value, 4.0);
            System.out.printf("%.3f^4 = %.3f\n", value, power);
        }
    }

    private static double getDouble(Scanner scnr)
    {
        double output;

        while (true)
        {
            System.out.print("Enter a double: ");
            try {
                output = scnr.nextDouble();
                break;
            } catch (InputMismatchException ex) {
                System.out.println("*** Error: please enter a double");
                scnr.next();
            }
        }

        return output;
    }
}
```

Figure 3: GetOutput.java

The following is an example session.

```
Enter a double: hello
*** Error: please enter a double
Enter a double: 33.214526252
33.215^4 = 1217060.730
```

Homework 3.2

4.1

In this question we're asked to determine which objects in the following pairs may be a subclass and parent class.

a. `Superhero`, `Superman` – in this case, `Superhero` would be the parent class and `Superman` the subclass.

b. `Justin`, `Person` – `Justin` is the subclass, and `Person` the parent class.

c. `Rover`, `Pet` – `Rover` is the subclass and `Pet` is the superclass.
d. `Magazine`, `Time` – `Time` is the subclass and `Magazine` is the superclass.
e. `Christmas`, `Holiday` – `Christmas` is the subclass and `Holiday` is the subclass.

4.4

Attributes I may include in a class `Course` to represent a college course include

- a list of instructors' names;
- the number of credits that can be earned;
- the course's name and related attributes, such as the course number;
- the start and end dates of the course (could be encoded in `Date`);
- location of the class;
- and a list of students.

Methods or operations I would write include

- a method to add or remove students from the course;
- a method to modify the location of the course;
- and a method to modify the number of credits a course may be.

4.5

In this question we're asked to write a method that has no return value and accepts no arguments that prints the lyrics of a song to the console. Please see my solution in Figure 4.

4.6

In this problem we're asked to write a method that accepts one integer parameter and returns the value raised to the third power. Please see Figure 5 for my solution.

4.10

In this problem we're asked to write the constructor of a class called `Movie` that initializes fields `name` and `director`. See Figure 6 for my solution.

4.11

In this question we're asked to write a getter and setter method for a field variable `age` in a class called `Child`. Please see my solution in Figure 7.

```java
package hw_3;

public class Lyrics
{
    public static void main(String[] args)
    {
        var lyrics = new Lyrics();
        lyrics.lyrics();
    }

    /**
     * Print the lyrics of a song.
     */
    public void lyrics()
    {
        // Source: http://www.chantcd.com/lyrics/hear_us_lord.htm
        // https://www.youtube.com/watch?v=L8FCpldsBV0
        System.out.println(
            "Hear us, O Lord, and have mercy, because we have sinned against Thee.\n"
          + "Hear us, O Lord, and have mercy, because we have sinned against Thee.\n"
          + "\n"
          + "To Thee, highest King,\n"
          + "Redeemer of all,\n"
          + "do we lift up our eyes\n"
          + "in weeping:"
          + "Hear, O Christ, the prayers\n"
          + "of your servants."
        );
    }
}
```

Figure 4: Lyrics.java

```java
package hw_3;

public class Cube
{
    public static void main(String[] args)
    {
        System.out.println((new Cube()).cube(5));   // 125
    }

    /**
     * Cube an integer.
     *
     * @param number The integer to be cubed.
     * @return The number cubed.
     */
    public int cube(int number)
    {
        return number * number * number;
    }
}
```

Figure 5: Cube.java

```
package hw_3;

public class Movie
{
    private String _title;
    private String _director;

    /**
     * Class constructor.
     *
     * @param title    Title of the movie.
     * @param director Name of the movie’s director.
     * @throws IllegalArgumentException If either title or director are empty
     *                                  strings.
     */
    public Movie(String title, String director)
            throws IllegalArgumentException
    {
        if (title.equals("") || director.equals(""))
            throw new IllegalArgumentException("Please provide valid arguments");
        this._title = title;
        this._director = director;
    }

    /* ... */
}
```

Figure 6: Movie.java

```
package hw_3;

public class Child
{
    private int _age;

    public Child(int age)
    {
        this._age = age;
    }

    /**
     * Our ‘getter‘ for _age.
     *
     * @return The age of the child.
     */
    public int getAge()
    {
        return this._age;
    }

    /**
     * Update the age stored in the child instance.
     *
     * @param newAge The child’s new age.
     */
    public void setAge(int newAge)
    {
        this._age = newAge;
    }
}
```

Figure 7: Child.java