CS 111, Programming Fundamentals II
Homework 4: Recursion

Computer Science

This homework is a bit different than the previous ones, it consists of 4 programming challenges. All of the programming challenges are about recursion. Refer to the lectures slides for examples and sample code. Recursion is very important in computer science, you will come across it again in upper level classes and throughout your studies. It is important that you solve these problems and understand how they work.

The code to each programming challenge is really short. The solutions to the first 3 programming challenges can be written in fewer than 10 lines of code, per challenge. Take your time and understand how it works, execute your code on paper, building the call stack and keeping track of the parameters. The solutions for the first 4 programming challenges can be put in the same class, name it **HomeworkFourChallanges**.

I. Programming Challenge 1 – Is it a Member

Write a recursive method named ***isItAMember***. The method should search an array of Strings for a specified String value, and return true if the value is found in the array, or false if the value is not found in the array. Demonstrate the method inside of your ***main*** method.

There is more than one way to set this up, basically you are given an array and you want to see if contains a certain element. Consider passing an additional parameter to the method which represents the size of the array.

II. Programming Challenge 2 – String Total Reverser

Write a recursive method named ***stringTotalReverser*** that accepts a string as its argument and prints the string in reverse character order and reverse capitalization of all characters. Demonstrate the method inside of your ***main*** method. Sample input/output can be seen below:

```
cwU123 -> 321uWC
CenTral -> LARtNEc
```

III. Programming Challenge 3 – An Interesting String

Write a method named ***interestingString*** that returns boolean value and uses recursion to determine whatever a String argument is indeed “interesting”. The method should return true if the argument reads the same forward and backward. Demonstrate the method inside of your ***main*** method. Below are a few examples of such strings.

anna	anna
kayak	kayak
radar	radar

IV. Programming Challenge 4 – Russian Dolls

Create a class called **Matreshka** (Russian doll). The fields should include: a ***name*** (String) and a ***height*** (double). In addition, there should be a field to hold Matreshka's ***daughter*** of type **Matreshka**. A constructor for the class can take a name and a height.

Matreshka's name can be a combination of the ***name*** and ***height***. For example, if Matreshka's name passed to the constructor is "Natasha", and the height is 5.0, the name can be "Natasha50" or "Natasha5.0" or "Natasha5". In the constructor matreshkas daughter will be created, with a height less than mothers by the value of 1. The smallest matreshka should have the height be 1 and its daughter should be null. Add getter methods for the instance fields as needed, for example to retrieve the "daughter" of the Matreshka, you can add a method ***getDaughter***, which returns a reference to the daughter.

Write a driver program for this programming challenge, another class with the ***main*** method. Write two additional methods in this class, the first one should be named ***decreasingHeight***. This method should reclusively print out matreshkas names in decreasing height order. The second method should be named ***increasingHeight***, which prints out the names in increasing height order. Both these methods should take in a reference to a **Matreshka**.

In the ***main*** method create a single instance of **Matreshka**, the name could be one of the following: Natasha, Olga, or Tatiana and the height can be a 9.0, 8.5, or 5.0. Call the methods ***decreasingHeight***, and ***increasingHeight***. Sample output can be seen below in **Figure 1**.

```
 ----jGRASP exec: java Driver
call to decreasingHeight()
natasha5.0
natasha4.0
natasha3.0
natasha2.0
natasha1.0

call to increasingHeight()
natasha1.0
natasha2.0
natasha3.0
natasha4.0
natasha5.0

 ----jGRASP: operation complete.
```

Figure 1 – Matreshka Sample Output

V. Upload your work to Canvas

Make sure that you upload the following files to the Homework 4 assignment in your Canvas account:

HomeworkFourChallanges.java
Matreshka.java
MatreshkaDriver.java
Screenshot image containing the output for MatreshkaDriver.java

Using the ***Snipping Tool*** take a screenshot of the output and save it as an image.

There will be additional files (the .class files, that you've generated when compiling your code) in your homework 4 folder, but don't upload them to Canvas.

V. Rubric

File / Lab	**Points**
Programming Challenge 1 – ***isItAMember*** solved with recursion	10
Programming Challenge 2 – ***stringTotalReverser*** solved with recursion	10
Programming Challenge 3 – ***interestingString*** solved with recursion	10
Programming Challenge 4 – **Matreshka** and **MatreshkaDriver** classes implemented correctly	35
Code is commented and nicely formatted and a screenshot is provided	5
Total	70