

# CSCI 1301 – Lab 27

November 27, 2019

## 1 A Guessing Game

We will revisit our guessing game from Lab 11<sup>1</sup>. If you haven't solved it already, have a look back and solve it.

Read carefully the challenge below, and think about the parts of your previous program you can re-use to solve this one.

Write a program that asks the user which level of difficulty (s)he wants to play: easy, medium or hard. Then, generate a random number: between 0 and 10 for easy, between 0 and 50 for medium, and between 0 and 100 for hard. Then, let the user try to guess the number you randomly generated:

- if the number given by the user is less than the random number generated, your program should display a message saying that the user should try with a greater value,
- if the number given by the user is greater than the random number generated, your program should display a message saying that the user should try with a lesser value,
- and if the number given by the user is the same as the random number generated, your program should display a message saying that the user won, and how many attempts it took him/her to guess correctly.

You can download a project that lays the foundations<sup>2</sup>, but it is recommended to start from scratch.

## 2 Wireless Service Provider

### 2.1 Description

A wireless service provider has three different subscription packages for its customers.

- Package A: For \$9.95 per month, 10 GB of data are provided. Additional GB are \$2.00 per GB.
- Package B: For \$14.95 per month, 20 GB of data are provided. Additional GB are \$1.00 per GB.
- Package C: For \$19.95 per month, unlimited data is provided.

We want to design and implement a class for this wireless service provider.

---

<sup>1</sup>../11/#a-guessing-game-optional

<sup>2</sup>RandomGuessGame.zip

## 2.2 Design

Design a class named `WSP` that calculates a customer's monthly bill. It should contains:

1. Attributes for both the letter of the package the customer has purchased (A, B, or C) and the amount of data (in GB) that the customer has used.
2. Accessors and mutators for all attributes.
3. A no-arg constructor that defaults to package A and no GB used.
4. A constructor that sets all fields using its parameters.
5. A method that calculates a customer's total charges for the package they have chosen.
6. A method that calculates and displays the amount of money Package A customers would have saved if they had purchased Package B or Package C, and the amount of money Package B customers would save if they purchased Package C.

Write the UML diagram for this class.

## 2.3 Implementation

Work on the `WSP` class and on a test program in parallel: implement the first three steps, and then write a small test program. Make sure you can compile your program, and that it behaves as expected before continuing. Write the second constructor, and test it by creating an object using it. Continue with the next-to-last step, test the method, and finally write the last step. To test this last method, you should create numerous objects, and call the method with them.

Try to use a `switch` statement for this last method.