CSCI 1301 – Lab 17

March 14, 2019

This lab will focus on getting you started with your second project. You can find the instructions for this project at http://spots.augusta.edu/caubert/teaching/2019/spring/csci1301/shared/project2.pdf. The hints below are relative to each problem, but I strongly recommend you go over them all no matter what problem you end up choosing.

1 "Daily Pay" Problem

This problem is about computing a value that partially depends on the user, and displaying it nicely.

1.1 Hint 1 - Computing

Observe the following code:

```
int a = 1;
int b = 10
while (b > 1){
    a = a + 2;
    Console.WriteLine($"a is now {a}.");
}
```

Adapt it so that

1. The loop actually stops!
2. The value of `a` would be *multiplied* by `2` insted of being *incremented* by `2` every time the loop execute.
3. The value of `b` would be set by the user.
4. The sum of all the values of `a` is computed and stored in a variable.

Once you completed all of those steps, your loop will be pretty close to what you actually need for this problem!

1.2 Hint 2 - Displaying

The other aspect of this problem is to display the values nicely. One way of doing this is by using the `String.Format` method. Observe the following:

```
Console.Write("*");
Console.Write(String.Format("{0, 10}", "Hello"));
Console.Write("*\n");
Console.Write("*");
Console.Write(String.Format("{0, -10}", "Hello"));
Console.Write("*\n");
```

Adapt it so that

1. The escape sequence `\t` is displayed before the `*\n` in both statements.
2. Instead of displaying "Hello", the value of a numerical variable is displayed.
3. The numerical value is displayed using the format specifier for money.

2 "Newspaper Machine" Problem

This program have two aspects: you need to loop ("*As long as the user did not gave me enough money nor decided to quit…*"), and to convert a character or string input (`Q`, for instance) into a number (0.25 in that case).

2.1 Hint 1 - Repeating

Write a program that asks the user for an integer, add that integer to all the previous integers entered so far, and repeat as long as the sum of all the integer entered so far is less than 100. A `do while` loop will be more appropriate for this task, something *like*:

```
1 int sum = 0;
2 do
3 {
4     Console.WriteLine("Enter an integer.");
5     // Here, we need some code to read from the user.
6     // And some code to sum the values entered so far.
7 } while (sum < 100);
```

2.2 Hint 2 - Converting the User Input

Now, in the problem, we don't want to ask the user for actual integers. We want to give the user the possibility to enter a character, and then to map that character to a value. Write a small program that

1. Declare a double variable, and a string variable.
2. Ask the user to enter "N", "D" or "Q", and assign the value entered to that string variable.
3. Assign 0.05 to the double if the string entered was "N", 0.10 if the string entered was "D", and 0.25 if the string entered was "Q".

3 "Rock, Paper, Scissors" Problem

For this problem, you need first to simulate how a computer could "play", and then to implement the logic of the game.

3.1 Hint 1 - Generating Random Numbers

For this problem, we need to be able to generate random numbers. This is the way we will simulate some really primitive artificial intelligence: the computer will play rock-paper-scissors by just randomly chosing one of the possibilities.

The code given in the problem (without the comments) is the following:

```
Random myRandomObject = new Random();
int a, i = 0;
while (i <= 100)
{
    a = myRandomObject.Next(1, 11);
    Console.Write(a + " ");
```

```
    i++;
}
```

1. Compile it, execute it, and observe what is displayed on the screen.
2. Are you able to modify this code, so that the values randomly generated are between 1 and 3?
3. Can you write a program that asks the user if (s)he wants to generate a new value, and display a new random number between 1 and 3 every time (s)he answers "Yes", and quit otherwise?

3.2 Hint 2 - The Logic of The Game

Complete the following table with **W**in, **L**oose or get a **T**ie:

If you play	and the computer plays	then you
Rock	**Rock**,	_____
Rock	**Paper**,	_____
Rock	**Scissors**,	_____
Paper	**Rock**,	_____
Paper	**Paper**,	_____
Paper	**Scissors**,	_____
Scissors	**Rock**,	_____
Scissors	**Paper**,	_____
Scissors	**Scissors**,	_____

Now, in our program, the user will enter a string made of a single character ("R", "S", "P") and the computer will randomly generate an integer (1, 2, 3), so we can't "directly" use this table in our program.

But, exactly as you associate a string made of only one character to a word (R $\leftrightarrows$ Rock, P $\leftrightarrows$ Paper, S $\leftrightarrows$ Scissors), you can associate an integer to a word, can't you? Then, how to decide who won or if it is a tie should be easy!

4 "Grade Calculator" Problem

This problem involves a bit of Mathematics that can be tricky to get correctly, and involve cleverly getting the information from the user.

4.1 Hint 1 - Do The Math!

The first thing you need to do for this problem is to understand the equation needed to compute the grade. Start by trying to reproduce the example given in the project description on paper. Can you get the same result? Can you find a way that seems simple and flexible to compute the current grade so far in all generality?

4.2 Hint 2 - Gathering The Data

Once you have the equation figured out, you need to understand how you can obtain the data from the user. You have two possible ways of doing it:

- Will you ask first for the number of quizzes taken, and then ask for the values, or

- Will you ask for the values, and take "-1", for instance, as a signal that this quiz did not happened yet.

Try to implement one or the other, without worrying about the computation at this point (just add, for instance, the value entered).