Lab 2.4 - Guessing Game

In this lab, you will use conditional statements and variables to build a simple number guessing game.

Content
Tools to aid in the development process

The most important steps in creating computer programs come before you even start to code. The planning and designing stages are important because they ensure that you carefully consider the program you are going to write, and that you carefully consider the skills and components that you have at your disposal, and the ones that you might need.

In order to help you in the development process, you will be provided with a planning worksheet. You have used one of these before in the course. This planning worksheet is a tool that you can use to help make the development process easier. If you fill in all components of the worksheet, then you will be better prepared to start coding.

Section 1 - I'm Thinking of a Number...

You will write a SNAP program to choose a random number between 1 and 10 and then ask the user to guess a number. If the user's guess matches the random number, the user wins. If not, the user loses. In either case, the user should be shown a message indicating whether he won or lost and the secret random number should be revealed.

1. Fill out a Planning Worksheet for the above program.

2. Write the simple version of the guessing game program described above.

3. Modify the program to keep asking the user for guesses until the correct number is given. Be sure to give a message after each guess, but only reveal the secret number when the user has guessed correctly and the game is over.

4. Add code to ask the player their name at the start of the game. Then, personalize the message for an incorrect guess by adding the player's name. For example, if Sarah is playing the game, then the message should say "Sorry, Sarah, that guess is not correct" instead of just "Sorry" when Sarah guesses incorrectly.

Big Ideas
Tools and technologies can be adapted for specific purposes

Now that you have created a basic number guessing game, it is time to create a new one with more advanced features. You can sort of consider this upgraded version as Guessing Game 2.0.

As you develop the upgraded game, don't just start with a blank program. Be sure to adapt your code from the first guessing game and alter it to include upgrades.

This is similar to how large companies provide upgrades to their software. They add features to preexisting versions. Occasionally they will start entirely from scratch, when a very important, new version of their software is require. But this is rare and they usually include quite a bit of code that was a part of the original version.

Section 2 - Game Upgrades

1. Modify your guessing game so that the player can decide the range of possible numbers from which the secret number can be chosen. After asking the player's name, ask what he or she wants the highest possible number to be. Then, instead of choosing a random number between 1 and 10, choose a random number between 1 and the number the player requested.

2. Add code to keep track of how many guesses the player has made. After the player guesses correctly, inform him or her how many tries it took before the correct number was guessed.

3. Increase the player's chances by telling him or her whether the guessed number is too high or too low instead of just that it is incorrect.

Grading Scheme/Rubric

Lab 2.4 Criteria	
1.1 Planning Worksheet	0.5 points
1.2 Simple version	0.25 points
1.3 Repeat till correct guess	0.25 points
1.4 User name	0.25 points
2.1 Range of numbers	0.25 points
2.1 Number of guesses	0.25 points
2.3 High or low	0.25 points

Lab 2.4 Criteria	
PROJECT TOTAL	2.0 points