

Table of Contents

Introduction.....	1
Summary.....	1
Details.....	2
void main().....	2
Game Setup.....	2
Game Loop.....	2

Introduction

This challenge assignment is optional and meant to replace certain assignment scores if you think that the usual assignment is below your level.

Summary

This Java program will begin running. It should generate a random number between 0 and 10. The player gets 5 guesses to guess the number.

Continue looping through the following code until either the player has guessed the correct number, or the player is out of guesses:

- Ask the player to guess a number
- Check to see if that number is greater, less than, or equal.
- If it is equal, the player has won. Stop looping and display "You Win!"
- Otherwise, tell the player if their guess is too high or too low.
- If the player is out of guesses, stop looping and display "You Lose!"

Details

You will create a class named **NumberGuesserProgram**. You will add methods within to flesh out the program's functionality.

void main()

This is the method that the user will run to begin the game.

Game Setup

To set up the game, you will need to generate a random number. Use Java's **Random** object to generate a number between **[1, 10]** (1 to 10, *including* 1 and 10.) Store this in a variable named **randomNumber**. It should be of type `int`.

Create another variable called **guessesRemaining**. It should be of type `int` and initialized to the value of 5, so that the player gets 5 guesses.

Game Loop

Create a game loop by using a **while loop** to continue running the program until the game is over.

The game is over either when:

- The amount of **guessesRemaining** is less than 0.
- The player has successfully guessed the random number.

You might want to create a **boolean** variable like “isRunning” or “isDone”, and have the while loop continue looping while **isRunning** is true or **isDone** is false. Then, within the loop you can set these to the appropriate value if an ending condition is hit. Alternatively, you may use **break;**, but you will have to look up how to use this.

Within the loop:

1. Ask the user to enter a number guess.
2. Get the user's input. Store it in an integer called **guess**. Use the **Scanner** class to get the user's input.
3. Check to see if the **guess** is correct – is it equal to the **randomNumber**?
 1. If the guess is correct:
 1. Display the message “You win!”
 2. Exit the loop.
 2. Otherwise, if the guess is not correct:
 1. Display a message that indicates whether the player guessed *too low* or *too high*.
 2. Decrease one from the **guessesRemaining** variable.
 3. (Then the game will loop again)

Once the loop has finished, display the message “Good bye!”. Once the end of the method is reached, the program will end.