The purpose of today’s lab is to develop a number guessing game.

The purpose of the game is to guess a number between 1 and 100. The lab is divided into two parts - Part 1 is a basic guessing game. If you get that finished, tackle Part 2 adds another game play option. To make this faster, note the code hints at the bottom of the lab page.

**Submit whatever you have at the end of the lab. No post lab submission.**

**Part 1. Guessing Game**

In this part, you’re creating a game that allows the user to guess the random number. The number entered is checked against the stored random number. If it’s higher or lower than the user guess again. If it’s correct, the user’s congrats message displays. Each guess accumulate an attempt.

A screenshot of a computer

AI-generated content may be incorrect.

Structure:

Two classes:

A GuessingGame class that contains :

**Attributes**

* An attribute of type int to store the random number generated (i.e. the number to be guessed);
* An attribute of type int to store the number of guesses.
* (An attribute to hold the Scanner object you will need to create – the need for this to be a class attribute will depend on how you structure your code);

**Constructor** – if you decide you need one, put one in.

**Methods to write:**

* A method to generate a number between 1 and 100;
* A play() method – that takes a guess in from the users, checks the number and does the interaction. Hint: You’ll need a [while](https://www.w3schools.com/java/java_while_loop.asp) loop that keeps looping till the guess is guessed
* A method to get user the user’s guess in from the console (and increment the number of guesses);

Add in any other methods you need to get it working.

A Control class that:

* Has a main method to run the game
* The main method should interact with the user
  + System.***out***.println("Choose a game mode:");

**Part 2. Hints Guessing Game**

If you have Part 1 working – you’ll now add a new version of the game so that users can pick to play the standard Guessing game from Part 1, or the Guessing Game.   
When the game starts, the user picks their choice of game.

A screenshot of a white background

AI-generated content may be incorrect.

You’ve already coded the Standard Number Guessing Game in Part 1. Now code up the Hinted Number guessing game which works as follows:   
  
When the user makes their guess, the game tells them the distance between their guess and the correct number, but not whether it’s plus or minus.

**Add a new class HintedGuessingGame** that inherits from (extends) GuessingGame

* It should have :
* A **play()** method, overriding the play() method from the GuessingGame – this may involve restructuring your guessinggame.java play() method a little.

Edit your Control class so that it allows the user to pick which game they want to play.

**Tip 1 -------To get user input from the console**

Use java’s Scanner class. This class, from the java.util package is used to get user input.

Scanner scanner = **new** Scanner(System.***in***); // set ups the console prompt to get user input;

String stringValueEntered **=** scanner.nextLine()); // takes the input from the user, and returns it as a string;

**int** integerValueEntered = Integer.*parseInt*(scanner.nextLine()); // takes the input from the user, and returns it as a integer (if an integer was entered, you need to convert it from the String type returned from by the scanner – you would normally use try/catch logic with this – but we’ll exclude that for now;

**Tip 2 -------To generate a random number between 0 and XXX**

Look up the Java API’s [Random class](https://docs.oracle.com/javase/8/docs/api/java/util/Random.html) – and find a suitable method allows you to generate a “ pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence”