

The "guessing game" is something that most of us have played as children. One person chooses a number, and the other person guesses until they get it correct. The person who knows the correct answer will give hints, "too low" or "too high", until the guessing player is correct.

There are several levels of difficulty to this assignment. To receive a high mark, all levels must be completed. Please complete all levels in order unless instructed otherwise.

Save your file as **Lastname_Firstname_Guessing_Game** and submit it to the appropriate folder.

This is an assignment - please work independently. In general, you should feel free to discuss the overall assignment and concepts, but students should not be helping each other with programming actual code. Students will be observed while working on the assignment and inappropriate interaction can affect your overall mark. In addition, all programs are checked for plagiarism.

Marks will be awarded in the categories of Application, Thinking and Communication.

Application – required features are present; features work properly

Thinking – efficiency of code and algorithms used to solve problem

Communication – quality of the user interface (instructions, prompts, messages to the user, overall appearance and formatting); quality of source code (program header, comments, good variable names, proper indenting, breaking up sections of code with blank lines for readability)

Note/Warnings:

1. Before you try another level, make sure you have perfected the previous level. It is better to do a great job at a lower level than a mediocre job with higher levels. Your overall mark will probably be better with a very good effort at the lower level.
2. Once you have a level working perfectly, save and submit the file before you make modifications. Some of the higher levels require extensive changes to the code, and you don't want to ruin code that was working (if you can't get the higher level working, you will have to hand in the lower level... better to have it ready to go!).
3. Create backup copies of your work! You can do this automatically by submitting to Edmodo every day, or even several times per day. Otherwise, you might use your own system, such as numbered or dated files or folders. For example,

```
programFile01.java  
programFile02.java  
programFile03.java
```

Level 2 (up to 70%) - meets minimum expectations for functionality

Create a program that:

1. provides a set of nicely formatted instructions on how to play the game
2. has the computer generate a random number between 1 and 100
3. the user is then asked to guess the number
4. if the user is wrong, they are informed whether their guess is too high or too low
5. the user is then allowed to guess again
6. this continues until the user guesses the number correctly (unlimited guesses)

Level 3 (up to 80%) - meets full expectations for this assignment

Instead of the given range of numbers, 1 to 100, the user can specify the range of numbers from which the computer must generate a random number (e.g., 10 to 50). That is, at the very beginning of the program the user will be asked to input the minimum and maximum numbers of the range. The computer will then generate the random number in this new range.

The program will also keep track of the number of guesses required to win the game. When the game is over, report the final number of guesses to the user.

Level 4 (up to 90%) - exceeds expectations for this assignment

Modify the program so that there is a limit on the number of guesses (for example: maximum of 10 guesses). If the user guesses the correct number before their attempts run out they win, otherwise the computer wins. There are better and worse ways to do this, and your effort will be graded accordingly.

Modify the program so that the computer keeps track of how long it takes the user to play the game and gives more points if the user is able to guess the number quickly. This feature may require you to do additional research into the programming language (e.g., how to measure and track time).

Level 4+ (up to 100%) - well beyond expectations for assignment

Add an option where the player and computer reverse their roles. In this modified game, the user will be asked to pick a number, and the computer will try to guess the number. For each computer guess, the user should indicate if the computer is too high, too low, or if the answer is correct.

An improvement to this would be to have the user input their number into the computer. You can trust the computer not to cheat (if you program it correctly).

As a final improvement, have the computer play both sides of the game against itself. All prompts and results should be shown just as if the computer were playing against a human.