

## CECS 277 – Lab 1 – Java Basics

### Guessing Game

Have the computer generate a number between 1 and 100, then prompt the user to input a number between 1 and 100. Test that the user's input is an integer and within the range, if it isn't, tell them that it is invalid and allow them to continue guessing. Compare the user's number to the computer's number, if the user's number is lower than the computer's, tell them that it is too low. If the number is higher than the computer's, tell them that it is too high. Repeat until the user guesses the number. Keep a record of how many tries it took and display it after the user wins. Do not count any invalid entries. Your output should be similar in appearance to the following.

**Example Output** (user input is in italics):

```
I'm thinking of a number. Guess a value (1-100): 10
Too Low.   Guess again: 50
Too High.  Guess again: 33
Too High.  Guess again: -14
Invalid.   Guess again: a
Invalid.   Guess again: 20
Too High.  Guess again: 15
Too Low.   Guess again: 17
Correct!   You got it in 6 tries.
```

### Notes:

1. You may use the CheckInput class provided on Beachboard to check user input for invalid values, just add the file to your project folder to use the functions. Feel free to modify it as needed. Examples using these functions is also provided on Beachboard.
2. Use Math.random() or the Random class to generate random numbers. Examples are provided on Beachboard.
3. Please read through Coding Standards on Beachboard for style guidelines. Modify your program as needed to conform to these standards.
4. Place your name, the date, and a brief description of the program in a comment block at the top of your program.
5. Add brief comments to your program to describe your code.
6. Thoroughly test your program before demoing.
  - a. Make sure the program accurately responds to high/low/correct guesses.
  - b. Count all valid guesses, including the final guess.
  - c. Do not count invalid guesses (ex. 0, negative numbers, values greater than 100, or any strings).
  - d. Invalid guesses should not output too high, or too low.
  - e. Make sure the program accurately reports the number of guesses the user made.