

Number Guessing Game

Write a guessing game program in which the computer generates a random integer number between 1 and 100, inclusive, and the player tries to guess that number. The program should allow a maximum of 7 guesses per game, and should tell the player whether a guess is too low, too high, or correct.



After a game is over, your program should score either the player or the computer 1 point depending on who wins the game, and then invite the player for another round of the game. This process should be repeated until the player chooses to stop playing the game. Finally your program should declare the player with a higher score the winner of the game.

Your program must perform input validation. If the user enters an invalid input value, your program should display an error message and then prompt the user to try again.

How to generate a random integer

To generate random numbers, you first create a `Random` object:

```
Random rand = new Random();
```

You can then call its `nextInt` method, passing it a maximum integer. The integer number returned will be between 0 (*inclusive*) and the maximum integer (*exclusive*):

```
int x = rand.nextInt(100); // 0 <= x <= 99
int y = rand.nextInt(100) + 1; // 1 <= y <= 100
```

Automate peek-before-read process

This assignment provides an excellent opportunity for you to start writing your own methods. To get you started, here is a method named `readNextInt` that takes a `Scanner` object as parameter and that *returns* an integer. Notice that the code in the method body is virtually the same code we have been using all along during a Peek/Read process. As you can see, the

code performs input validation before reading a number. It keeps prompting the user until the user enters an integer which it finally returns. So no more repeating the Peek/Read code!

```
public static int readNextInt(Scanner scan)
{
    while (!scan.hasNextInt())
    {
        String badInput = scan.next(); // consume the input
        System.out.print("Bad input '" + badInput +
            "' . Please enter an integer.");
    }
    return scan.nextInt();
}
```

Here is an example of how to use the method above:

```
// a sample code fragment
Scanner keyboard = new Scanner(System.in);
int guess;
// ...
System.out.println("I'm thinking of a number.");
System.out.print("Guess what it is: ");
guess = readNextInt(keyboard);
```

Formatting Your Output

A sample run of your program should look like this:

```
Welcome to my Number Guessing Game!
```

```
RULES:
```

1. We will play as many rounds as you wish.
2. Each round you will guess the number I've picked.
3. If you guess the correct value in 7 or fewer tries then you score a point, otherwise I score a point.
4. Whoever has the most points after all rounds wins.

ROUND 1

I've picked an integer from 1 to 100. 27 <- only during program tetsing

Guess it: 27

You win this round! You got it right in 1 guess

Another round (y/n)? y

ROUND 2

I've picked an integer from 1 to 100. 42 <- only during program tetsing

Guess it: 50

Guess lower: 25

Guess higer: 37

Guess higer: 44

Guess lower: 41

Guess higer: 43

Guess lower: Last chance! 42

You win this round! You got it right in 7 guesses

Another round (y/n)? y

ROUND 3

I've picked an integer from 1 to 100. 5 <- only during program tetsing

Guess it: 777

Your guess is out of range- guess again: -555

Your guess is out of range- guess again: 5

You win this round! You got it right in 1 guess

Another round (y/n)? y

ROUND 4

I've picked an integer from 1 to 100. 63 <- only during program tetsing

Guess it: 50

Guess higer: 75

Guess lower: 70

Guess lower: 65

Guess lower: 60

Guess higer: 61

Guess higer: Last chance! 62

You lose this round! I was thinking of the number 63

```
Another round (y/n)? n
Scores:
-----
  You: 3
  Me : 1

*** You Win! ***
Thanks for playing my Number Guessing Game!
```

Here is another sample run of the program:

```
Welcome to my Number Guessing Game!

RULES:
1. We will play as many rounds as you wish.
2. Each round you will guess the number I've picked.
3. If you guess the correct value in 7 or fewer tries
   then you score a point, otherwise I score a point.
4. Whoever has the most points after all rounds wins.

ROUND 1
-----
I've picked an integer from 1 to 100. 60 <- only during program tetsing
Guess it: 50
Guess higer: -10
Your guess is out of range- guess again: 777
Your guess is out of range- guess again: 60
You win this round! You got it right in 2 guesses

Another round (y/n)? y

ROUND 2
-----
I've picked an integer from 1 to 100. 19 <- only during program tetsing
Guess it: 100
Guess lower: 90
Guess lower: 80
Guess lower: 70
Guess lower: 60
Guess lower: 50
Guess lower: Last chance! 40
```

```
You lose this round! I was thinking of the number 19

Another round (y/n)? n
Scores:
-----
    You: 1
    Me  : 1

*** We tied ***
Thanks for playing my Number Guessing Game!
```

Evaluation Criteria

Correctness of execution of your program	60%
Proper use of required Java concepts	20%
Java API documentation style	10%
Comments on nontrivial steps in code, Choice of meaningful variable names, Indentation and readability of program	10%