Java Programming 1

PROFESSOR: CÂI FILIAULT

Topics for this week

- Repetition Structures
- Sentinel Values
- While loop
- For Loop
- Do While Loop
- Nested Loops
- Boolean Conditions

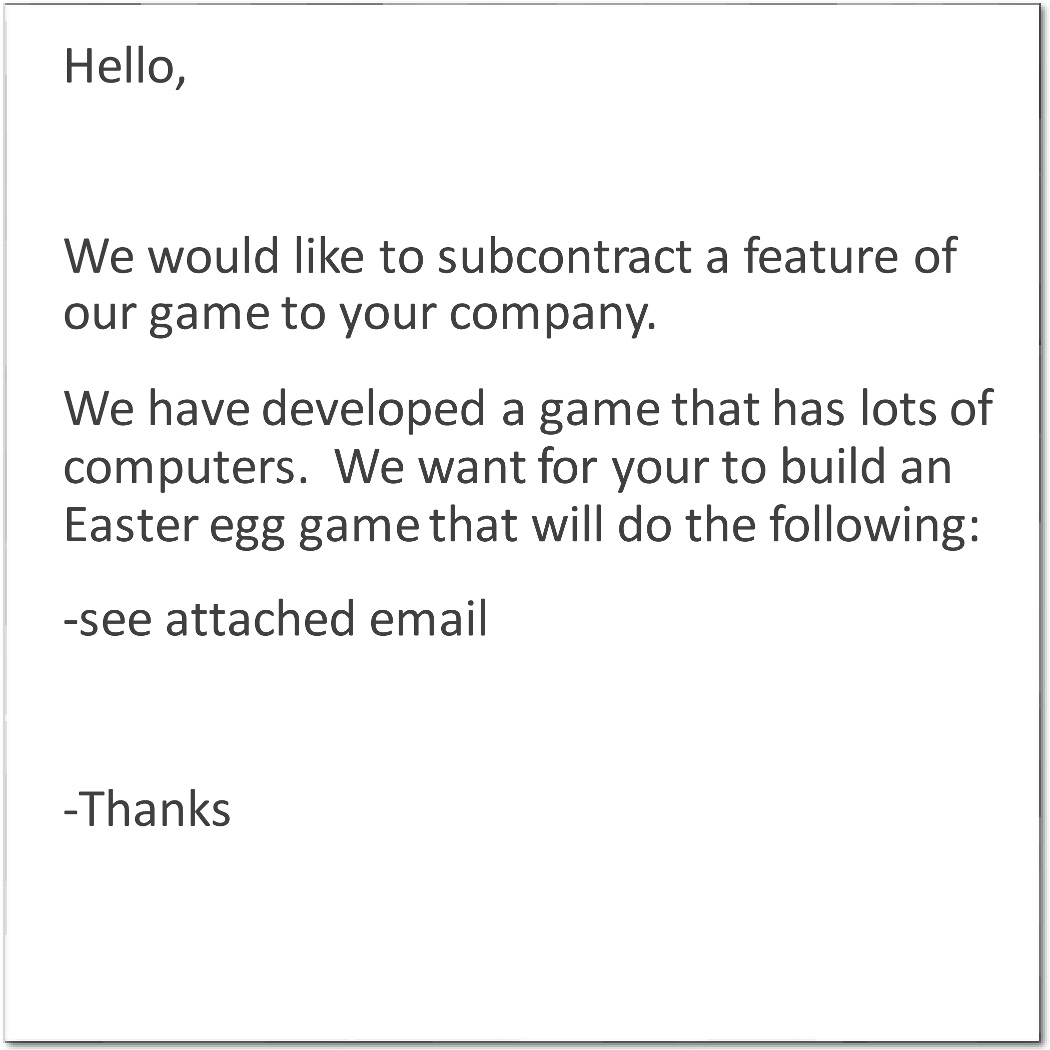
Record Games

GAME DEV. COMPANY

Email

Marty A. Theodore

School Teacher



Hello,

We would like to subcontract a feature of our game to your company.

We have developed a game that has lots of computers. We want for your to build an Easter egg game that will do the following:

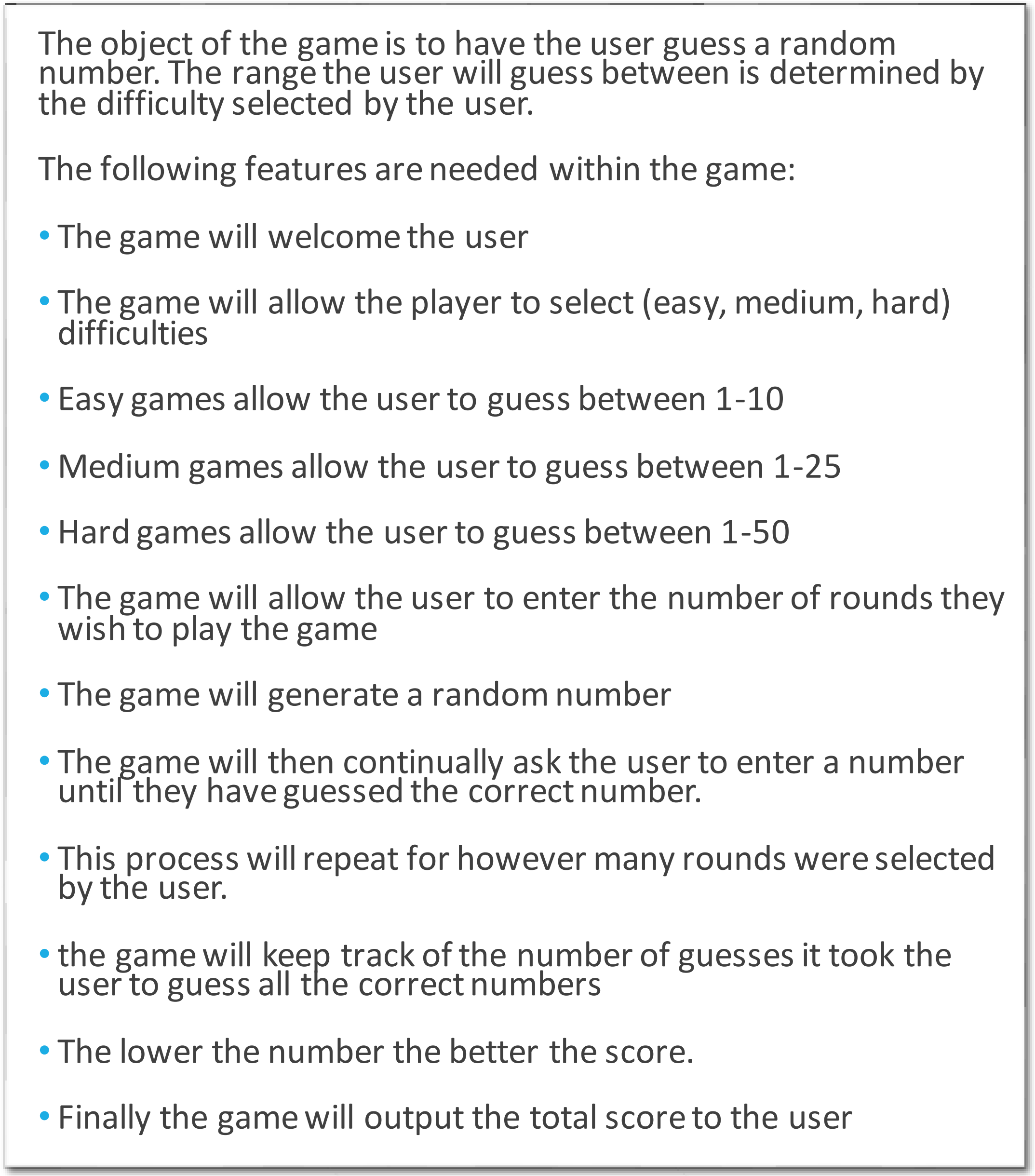
-see attached email

-Thanks

Email

Marty A. Theodore

School Teacher



The object of the game is to have the user guess a random number. The range the user will guess between is determined by the difficulty selected by the user.

The following features are needed within the game:

- The game will welcome the user
- The game will allow the player to select (easy, medium, hard) difficulties
- Easy games allow the user to guess between 1-10
- Medium games allow the user to guess between 1-25
- Hard games allow the user to guess between 1-50
- The game will allow the user to enter the number of rounds they wish to play the game
- The game will generate a random number
- The game will then continually ask the user to enter a number until they have guessed the correct number.
- This process will repeat for however many rounds were selected by the user.
- the game will keep track of the number of guesses it took the user to guess all the correct numbers
- The lower the number the better the score.
- Finally the game will output the total score to the user

Plan of Action

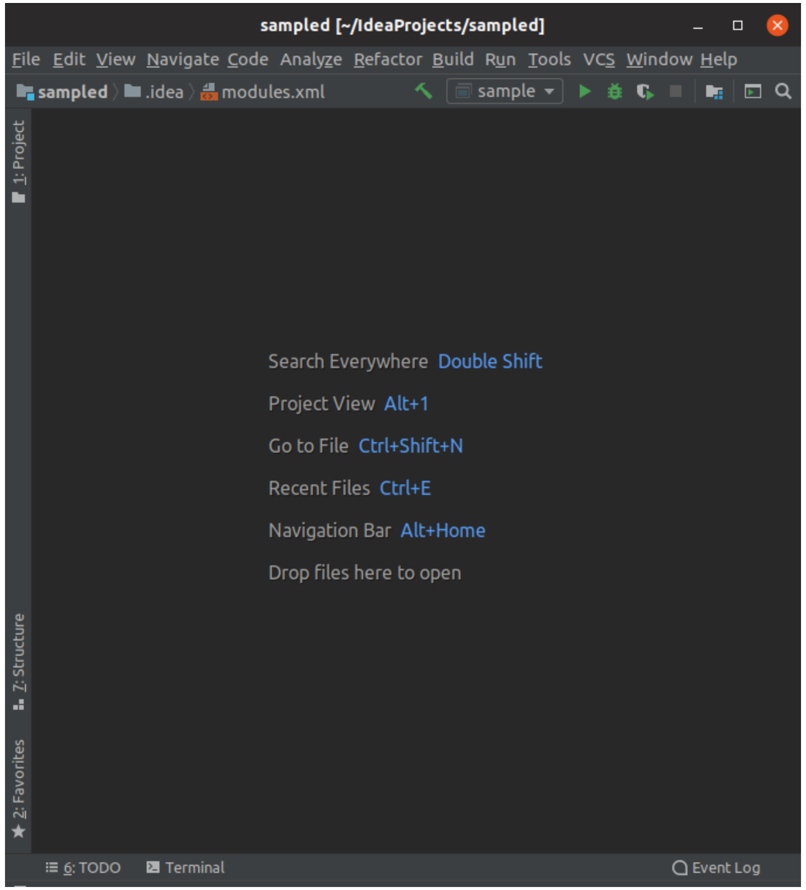
We will be generating random integers based on a value from 1-10, 1-25, 1-50

- We will need to learn how to generate truly random integers
- We will need to learn about repetition structures so that we can continually ask the user to input information
- We will need to utilize repetition structures to control the amount of rounds the user inputs information

Creating a new project

Start by opening IntelliJ and selecting the following:

File> New> Project



New Project

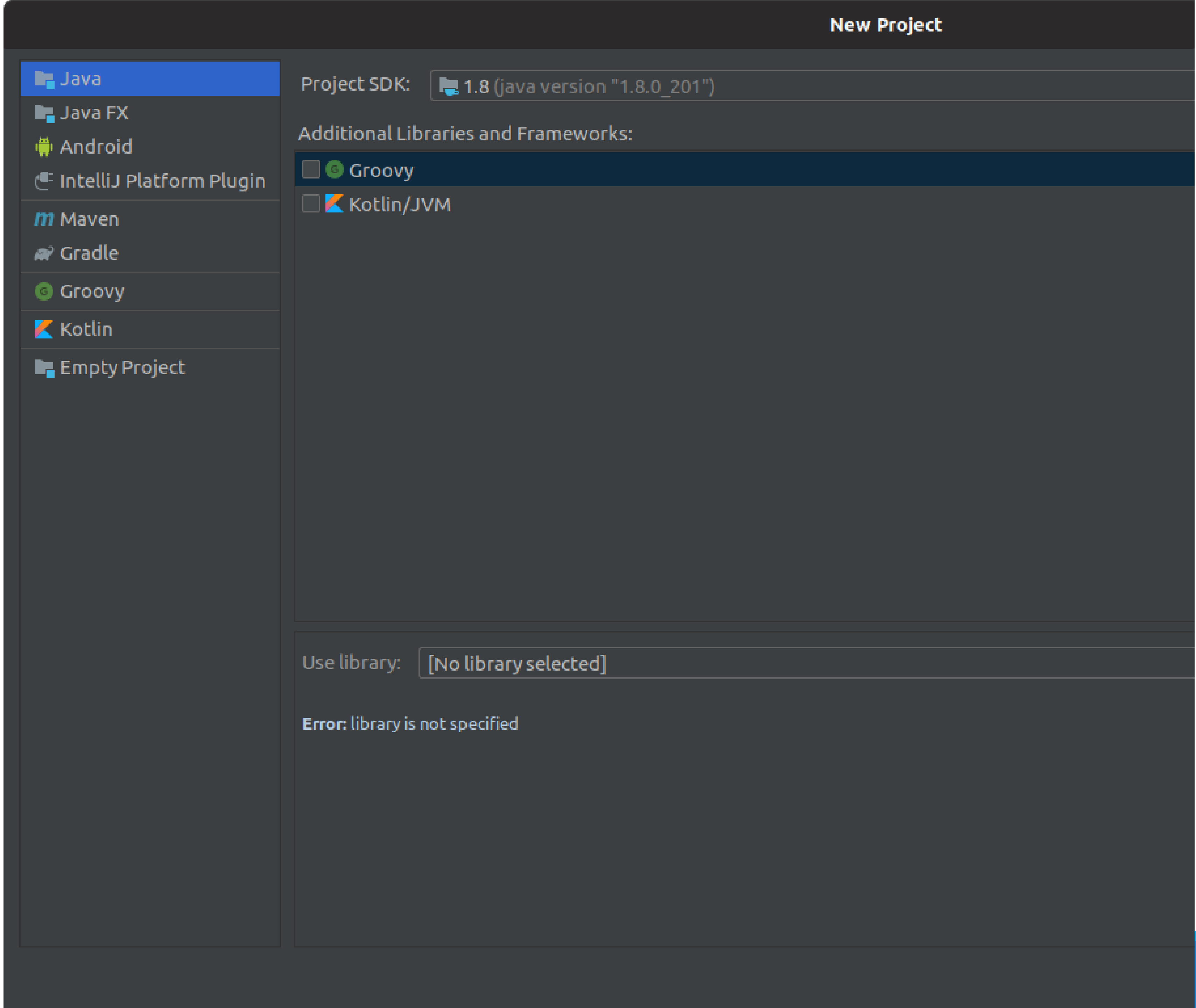
You will now see a dialog box appear *(It may be a little different than the one that I have shown)*

Starting on the left we will select a new Java Project

We will now want to select the project SDK of 1.8

SDK stands for Software Development Kit and it dictates which version of Java we are using.

At this point we can hit the next button twice



Project SDK: 1.8 (java version "1.8.0_201")

Additional Libraries and Frameworks:

- □ Groovy
- □ Kotlin/JVM

Use library: [No library selected]

Error: library is not specified

New Project

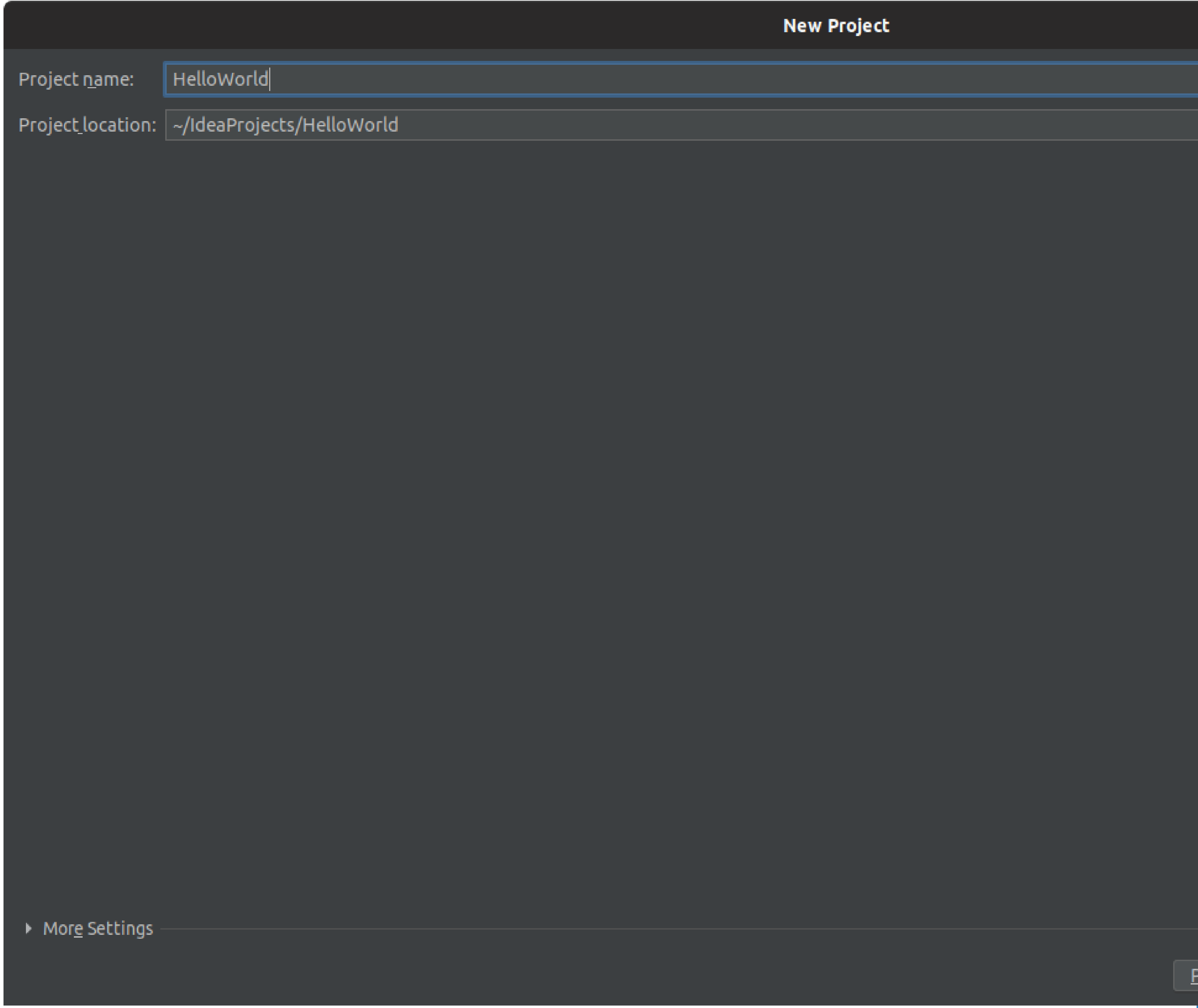
Give the project a name of:

RandomGame

At this time you can change the project location to any convenient place you would like

Tip:

Keeping your programming projects on a flash/jump drive is perfectly fine. However you will want to avoid from directly working on the drive.



New Project

Project name: HelloWorld

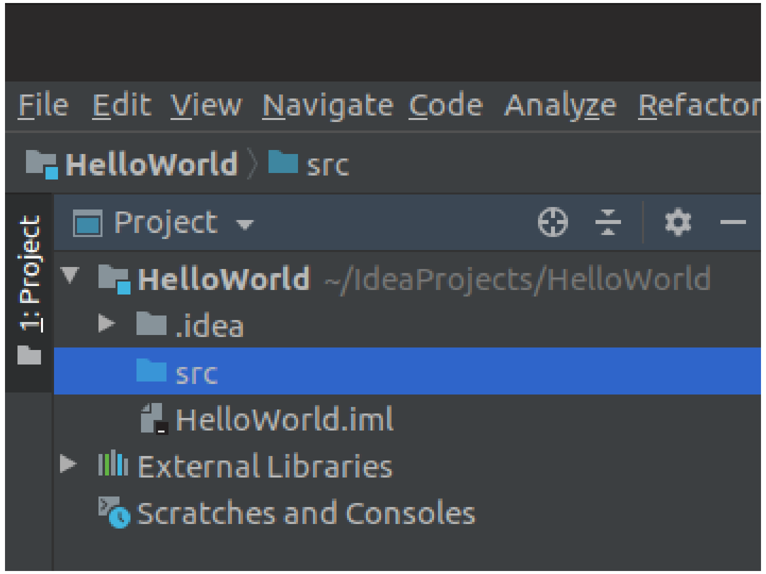
Project location: ~/IdeaProjects/HelloWorld

Creating a new project

You will now see in the top left-hand corner of your IDE a package explorer.

Within the package explorer you should see the **RandomGame** project.

Expand the **RandomGame** project to see a folder named **src**.



Creating a new class file

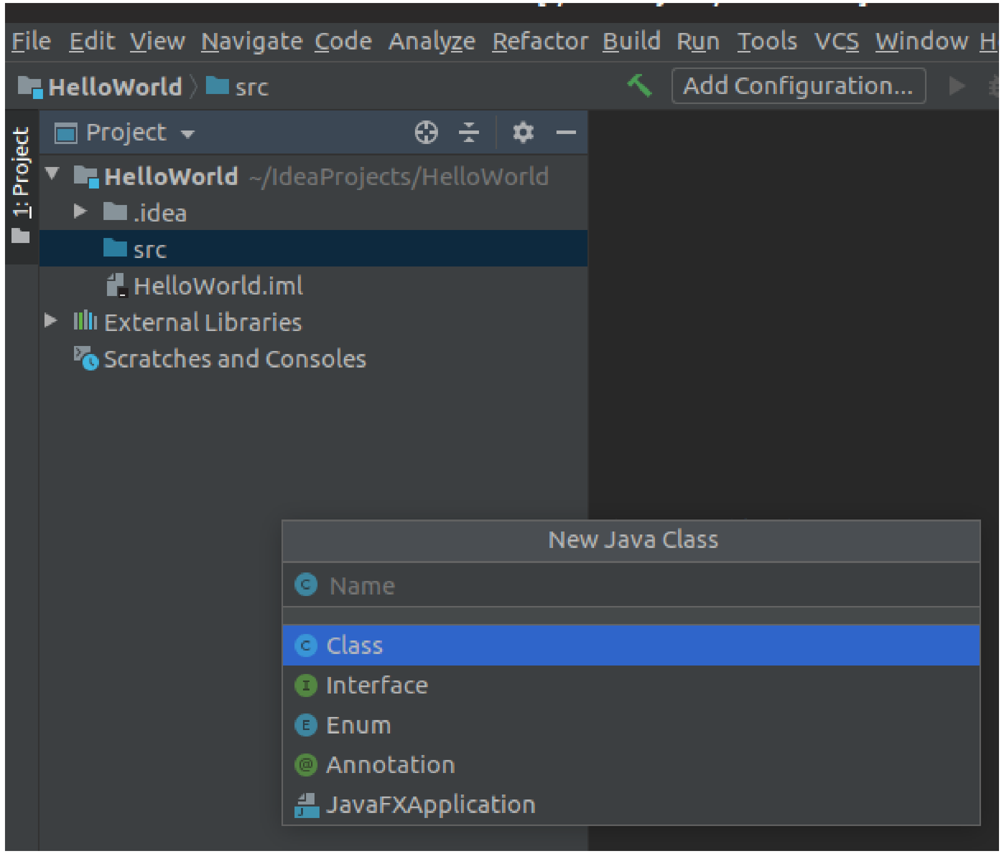
Next, we will need to create a new class file.

Right click on the **src** folder and select:

New> Java Class

Creating a new class is like creating a new program

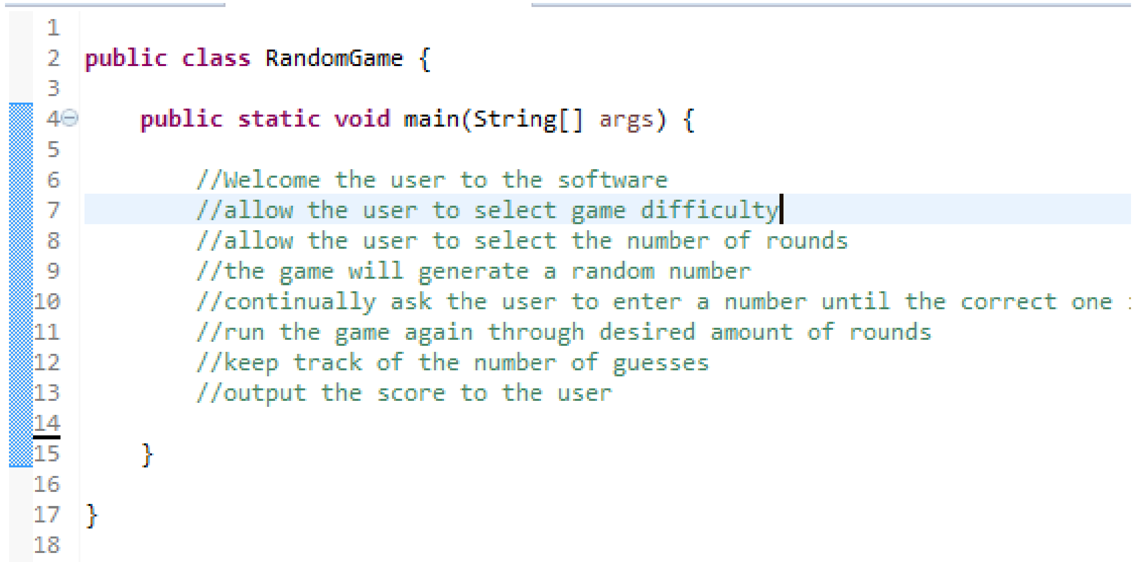
Name the class **RandomGame** and hit **enter**



Writing the program

You should now see a RandomGame.java file opened on the screen.

- Refer to the image on the right to see what your file should look like
- The first thing we are going to do is type "psvm" and hit the enter key
- This is what's known as "auto-completion"
- Add the comments to match ours shown

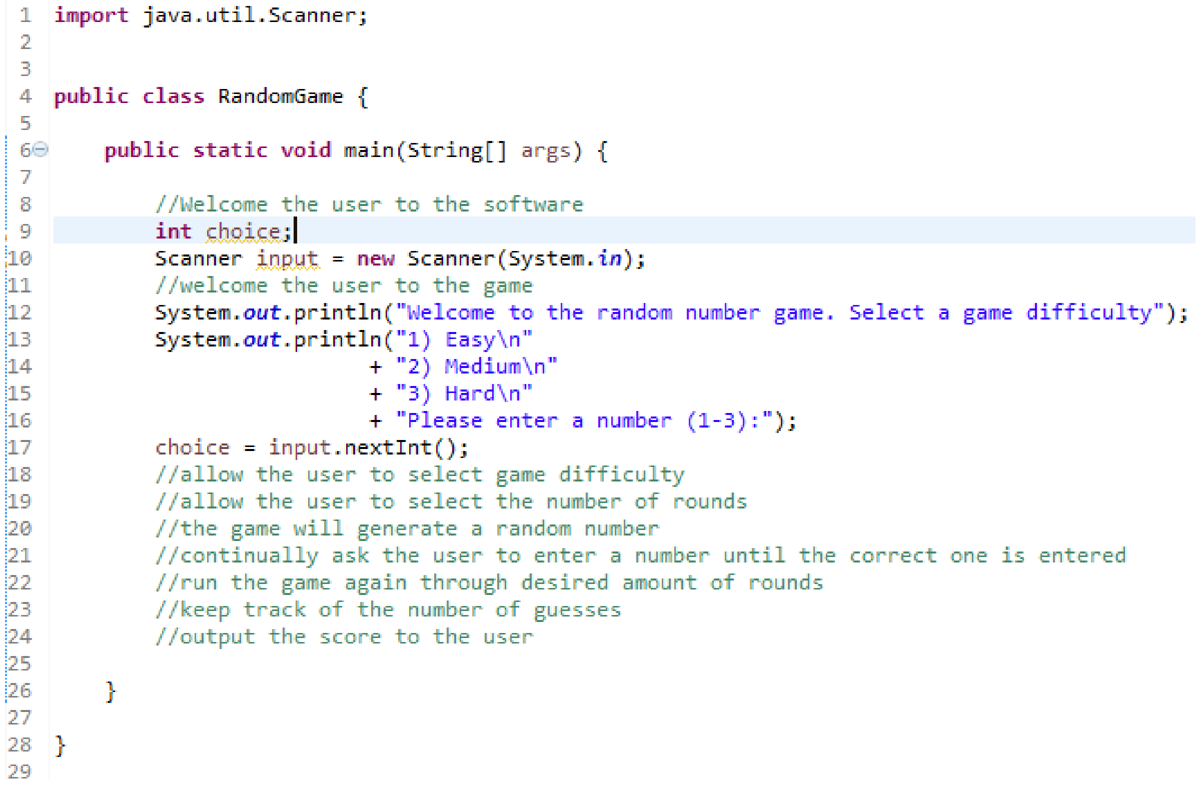


```

public class RandomGame {

    public static void main(String[] args) {

        //Welcome the user to the software
        //allow the user to select game difficulty
        //allow the user to select the number of rounds
        //the game will generate a random number
        //continually ask the user to enter a number until the correct one
        //run the game again through desired amount of rounds
        //keep track of the number of guesses
        //output the score to the user

    }

}

```

Writing the program

Line number 12- 16 introduce the user:

- Introduce the user to the program by adding the System.out.println as seen in line number 12 and 13
- We prompt the user to enter the difficulty they would like to play the game
- We also created a scanner class that is utilized to grab input from the user.
- We then grab input from the user (the game difficulty they would like to select)

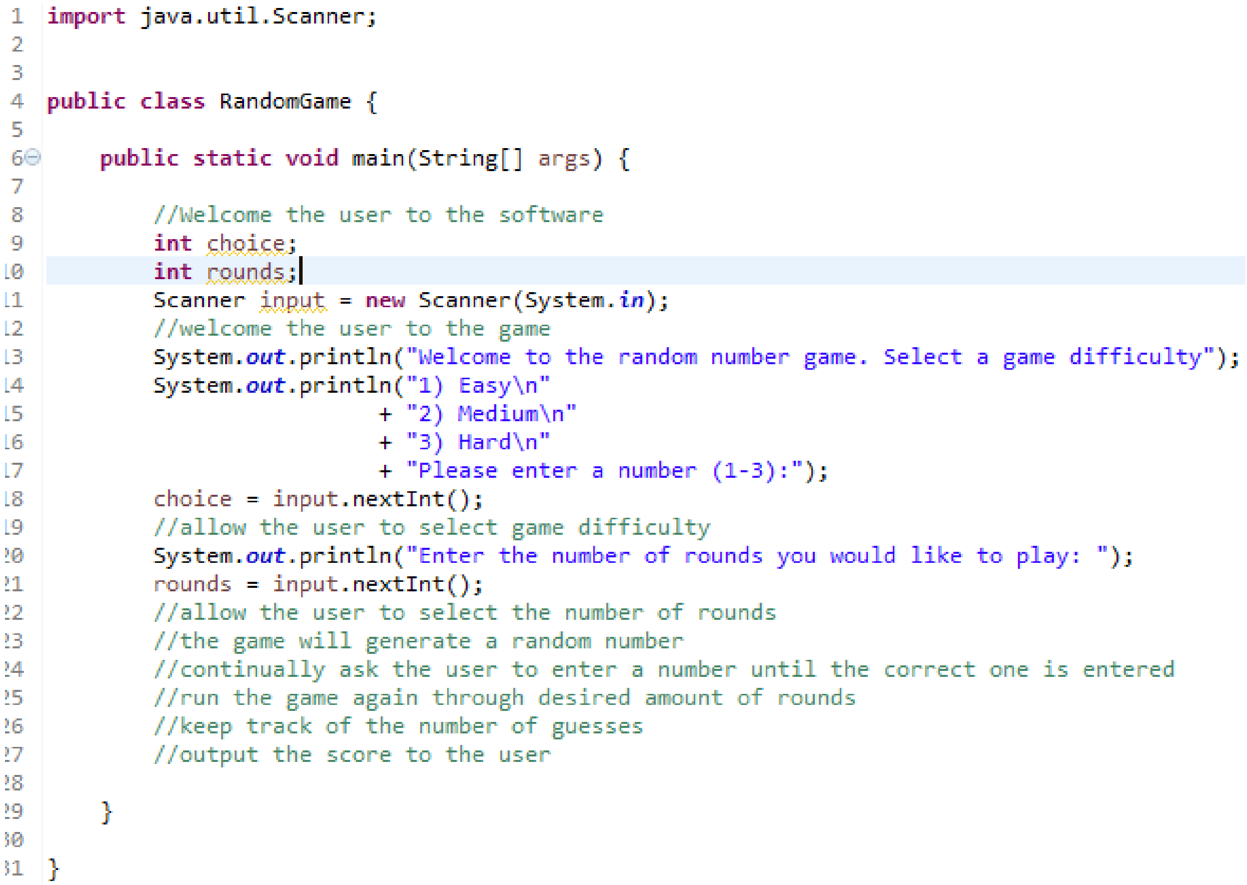


```
 1  import java.util.Scanner;
 2
 3
 4  public class RandomGame {
 5
 6      public static void main(String[] args) {
 7
 8          //Welcome the user to the software
 9          int choice;
10          Scanner input = new Scanner(System.in);
11          //welcome the user to the game
12          System.out.println("Welcome to the random number game. Select a game difficulty");
13          System.out.println("1) Easy\n"
14                          + "2) Medium\n"
15                          + "3) Hard\n"
16                          + "Please enter a number (1-3):");
17          choice = input.nextInt();
18          //allow the user to select game difficulty
19          //allow the user to select the number of rounds
20          //the game will generate a random number
21          //continually ask the user to enter a number until the correct one is entered
22          //run the game again through desired amount of rounds
23          //keep track of the number of guesses
24          //output the score to the user
25
26      }
27
28  }
29
```

Writing the program

Gathering Inputs:

- The next step is to gather the number of rounds the user would like to play the game
- We do this through prompting the user to enter the number of rounds on line 10
- We then utilize the Scanner class previously created to grab an integer number from the user on line 11
- The next step is to generate a random number
- Before we generate a random number, we need to learn a new concept



```
1  import java.util.Scanner;
2
3
4  public class RandomGame {
5
6      public static void main(String[] args) {
7
8          //Welcome the user to the software
9          int choice;
10         int rounds;
11         Scanner input = new Scanner(System.in);
12         //welcome the user to the game
13         System.out.println("Welcome to the random number game. Select a game difficulty");
14         System.out.println("1) Easy\n"
15                         + "2) Medium\n"
16                         + "3) Hard\n"
17                         + "Please enter a number (1-3):");
18         choice = input.nextInt();
19         //allow the user to select game difficulty
20         System.out.println("Enter the number of rounds you would like to play: ");
21         rounds = input.nextInt();
22         //allow the user to select the number of rounds
23         //the game will generate a random number
24         //continually ask the user to enter a number until the correct one is entered
25         //run the game again through desired amount of rounds
26         //keep track of the number of guesses
27         //output the score to the user
28
29     }
30
31 }
```

Generating Random Numbers

To generate a random number:

- Generating a random number is almost never random.
- Most random numbers require some sort of seed to determine its "randomness"
- The concept of seeding is utilizing seemingly unique or random information at compile time to generate a random number.
- In the case of Java the easiest seed to use is the current time in milliseconds.
- The likelihood of a user running the software at the exact same time is very low.
- A popular book by Kevin Mitnick & WIlliam L. Simon named "The art of intrusion" talks about how a casino got exploited for not using random enough numbers.

https://www.amazon.ca/Art-Intrusion-Exploits-Intruders-Deceivers/dp/0471782661

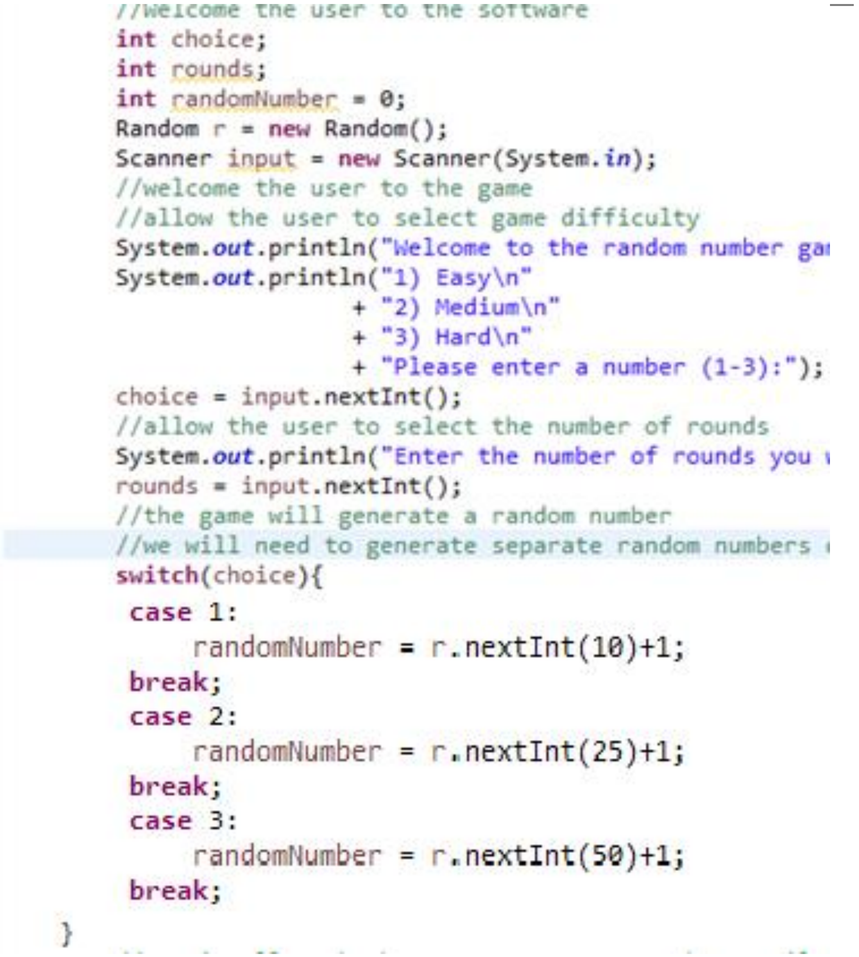
Generating Random Numbers

Demo of seeding

Writing the program

Let's look at how we will generate random numbers:

- **Line number 13** creates a new random object and stores it within a variable named r.
- Random() is a construct that creates a random object. For th sake of example let's call this a random machine.
- At the time of creation the random machine will take the exa time in milliseconds and use it as a seed when producing its numbers.
- **Line numbers 30, 34, 38** creates a random number utilizing t random machine, we tell it we want a number between 0 – 1 0 – 25, 0 – 50
- We need to add one to this number because we are expectir results between 1 – 10, 1 – 25. 1 - 50



```
//welcome the user to the software
int choice;
int rounds;
int randomNumber = 0;
Random r = new Random();
Scanner input = new Scanner(System.in);
//welcome the user to the game
//allow the user to select game difficulty
System.out.println("Welcome to the random number ga
System.out.println("1) Easy\n"
                  + "2) Medium\n"
                  + "3) Hard\n"
                  + "Please enter a number (1-3):");
choice = input.nextInt();
//allow the user to select the number of rounds
System.out.println("Enter the number of rounds you
rounds = input.nextInt();
//the game will generate a random number
//we will need to generate separate random numbers
switch(choice){
 case 1:
     randomNumber = r.nextInt(10)+1;
 break;
 case 2:
     randomNumber = r.nextInt(25)+1;
 break;
 case 3:
     randomNumber = r.nextInt(50)+1;
 break;
}
```

Writing the program

The next step asks us to ask the user to continually enter a number until they have guessed the correct number.

Repetition structures allow us to repeat a code until a certain condition is met. The main types are as follows:

- For Loop
- While Loop
- Do While Loop

Writing the program

For Loop

- For loops are used to execute blocks of code a set amount of times
- The for loops is comprised of:
  - A sentinel value (in the following example this will be known as i)
  - A condition (this will determine when we are to exit the for loop)
  - A way to manipulate the sentinel value so that it meets the condition
- For loops are used for a pre-determined amount of time
- For loops are pre check repetition structures, where they check the condition before executing the code

Writing the program

Do While Loop

Do While loops are used to execute blocks of code an undetermined amount of time (but at least once)

The do while loop is comprised of:

- A Boolean condition (this will determine when we are to exit the while loop)
- A way to manipulate the sentinel value so that it meets the condition

While loops are post check repetition structures where the condition will be checked after the code is executed at least once

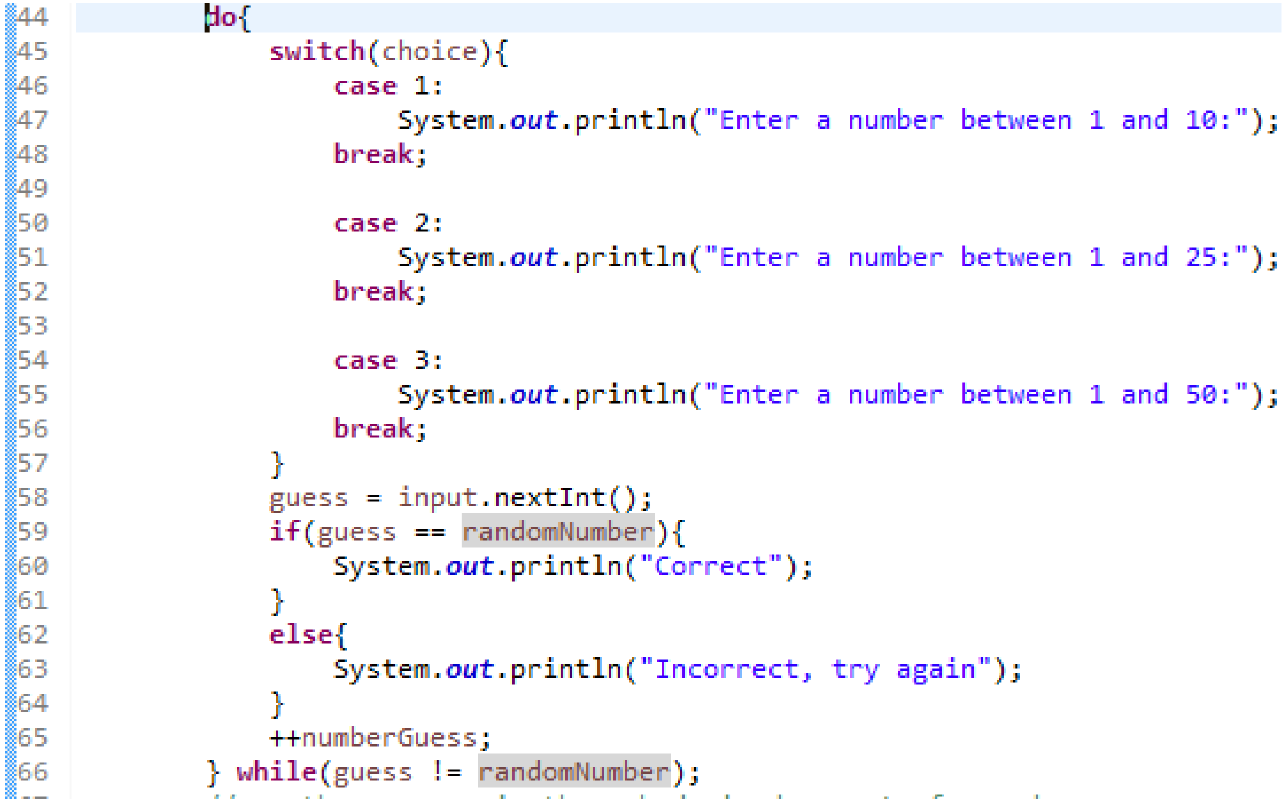
Do While loops are great for executing a block of code before deciding if you want to execute it again

What repetition structure would be good continuously prompting for input until the user enters the correct number?

Writing the program

Line number 44 creates the opening of a do while loop, once the block is opened all code within the block will be executed an undetermined amount of time (until the while loop is no longer satisfied)

Line number 66 ends the do while block and gives the condition for executing the code again.



```
44         do{
45             switch(choice){
46                 case 1:
47                     System.out.println("Enter a number between 1 and 10:");
48                 break;
49 
50                 case 2:
51                     System.out.println("Enter a number between 1 and 25:");
52                 break;
53 
54                 case 3:
55                     System.out.println("Enter a number between 1 and 50:");
56                 break;
57             }
58             guess = input.nextInt();
59             if(guess == randomNumber){
60                 System.out.println("Correct");
61             }
62             else{
63                 System.out.println("Incorrect, try again");
64             }
65             ++numberGuess;
66         } while(guess != randomNumber);
```

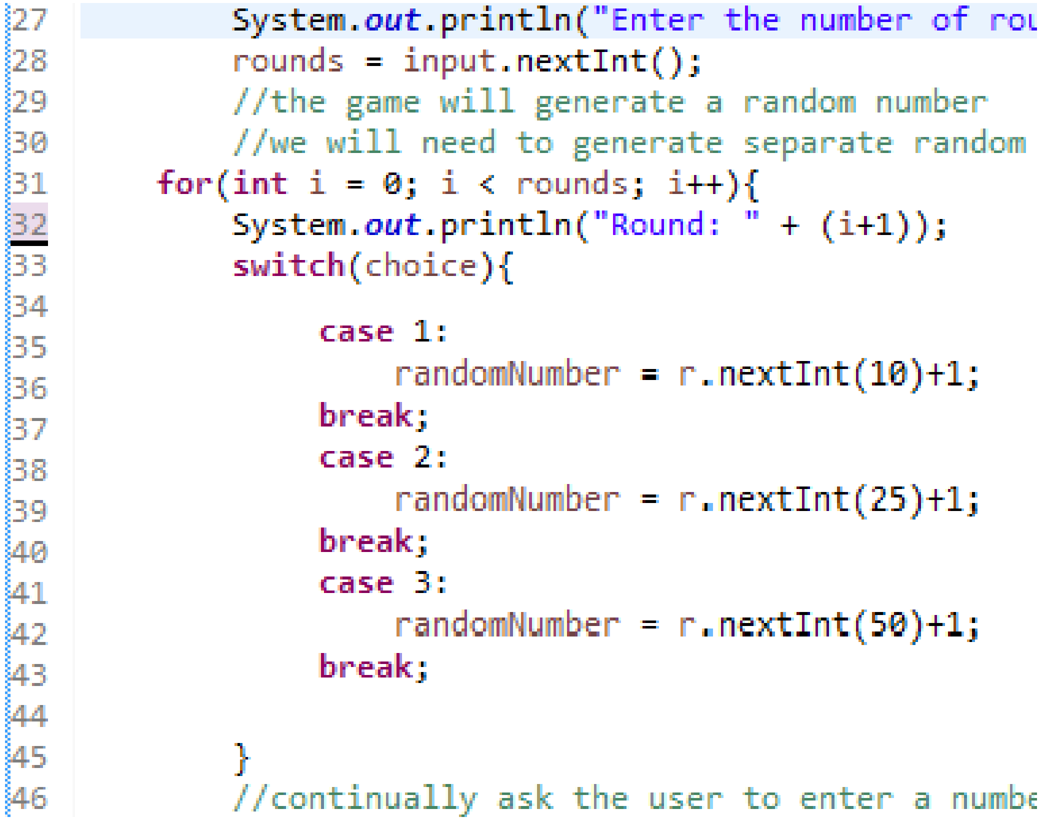
Writing the program

The next step is to run the program again a determined amount of time

- Recall we asked the user to enter the amount of rounds they wish to play the game
- We stored this information in a variable named rounds
- We will utilize this variable and a for loop to execute the program over again (equal to the amount of rounds determined by the user)
- We will also output to the screen what round the user is in

Writing the program

- **Line number 31** shows us creating a new for loop with a sentinel value i (as mentioned earlier)
- The condition is that we continually run the following block of code while i < the number of rounds entered by the user.
- We then increment the sentinel value by 1 so that it will eventually no longer meet the condition previously defined.
- **Line 32** we output the current value of i + 1 so that we output the correct round number they are in.

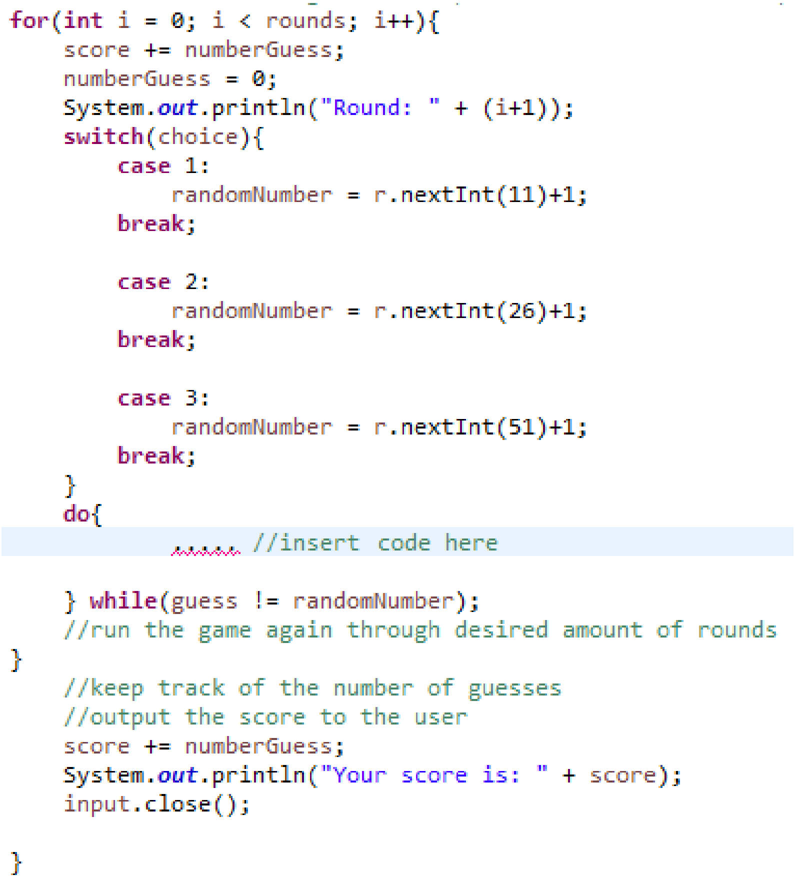


```
27          System.out.println("Enter the number of rou
28          rounds = input.nextInt();
29          //the game will generate a random number
30          //we will need to generate separate random
31      for(int i = 0; i < rounds; i++){
32          System.out.println("Round: " + (i+1));
33          switch(choice){
34 
35              case 1:
36                  randomNumber = r.nextInt(10)+1;
37              break;
38              case 2:
39                  randomNumber = r.nextInt(25)+1;
40              break;
41              case 3:
42                  randomNumber = r.nextInt(50)+1;
43              break;
44 
45          }
46          //continually ask the user to enter a numbe
```

Writing the program

We can keep track of the score by:

- Creating a local variable named score and making it equal to itself plus the number of turns it took to complete the previous round



```
for(int i = 0; i < rounds; i++){
    score += numberGuess;
    numberGuess = 0;
    System.out.println("Round: " + (i+1));
    switch(choice){
        case 1:
            randomNumber = r.nextInt(11)+1;
        break;

        case 2:
            randomNumber = r.nextInt(26)+1;
        break;

        case 3:
            randomNumber = r.nextInt(51)+1;
        break;
    }
    do{
            ...... //insert code here

    } while(guess != randomNumber);
    //run the game again through desired amount of rounds
}
    //keep track of the number of guesses
    //output the score to the user
    score += numberGuess;
    System.out.println("Your score is: " + score);
    input.close();

}
```

Test your software

Take a moment to test your software

Homework

Read Chapter 5 of your textbook

Next Week

- Repetition Structures
- Sentinel values
- While loop
- For loop
- Do While Loop
- Nested for loops
- Nested loops
- Boolean conditions
- Character and String manipulation