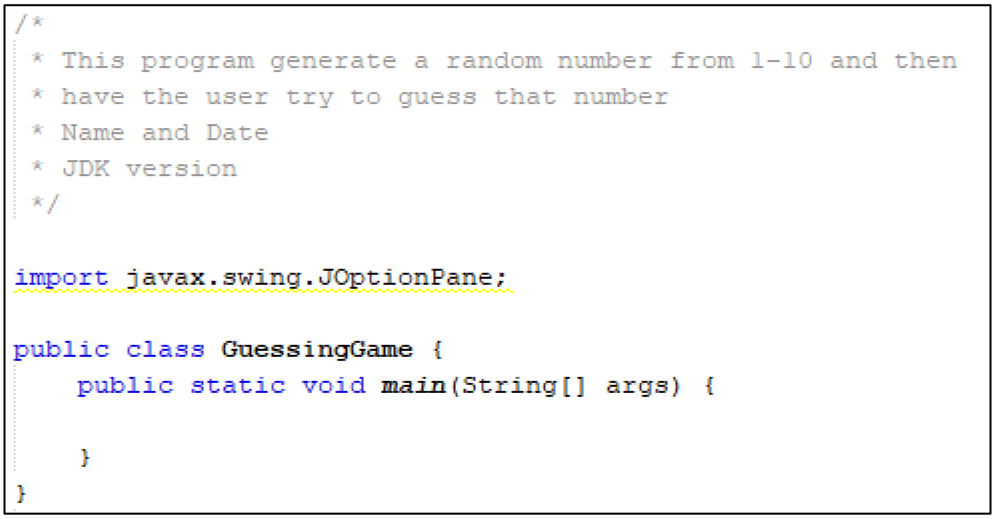
Guessing Game Exercise

The computer generates a random number between 1 and 100 and asks the user to guess the number. The computer will then respond with messages depending upon the user's answer.

1. Get into NetBeans and start a new Java program. The name of the project and main class file should be: **GuessingGame.** Type in the following portion of the Java program.



```
/*
 * This program generate a random number from 1-10 and then
 * have the user try to guess that number
 * Name and Date
 * JDK version
 */

import javax.swing.JOptionPane;

public class GuessingGame {
    public static void main(String[] args) {

    }
}
```

2. The first task is for the computer to generate a random number between 1 and 100. There is a method called random in the Math class that generates a random double number approximately between .0000000000 and .9999999999.

3. If we simply have the statement of :

   int computerNumber = Math.random( );

   We will get a compiler error. Do you have any idea why? The problem is that Math.random() returns a double number and then you are trying to put it in an int variable of computerNumber. Java will require us to cast this double to an int. So now the line would be:

   int computerNumber = (int) (Math.random( ));

   This will make the random number generated an int by dropping off the decimal part of the number.

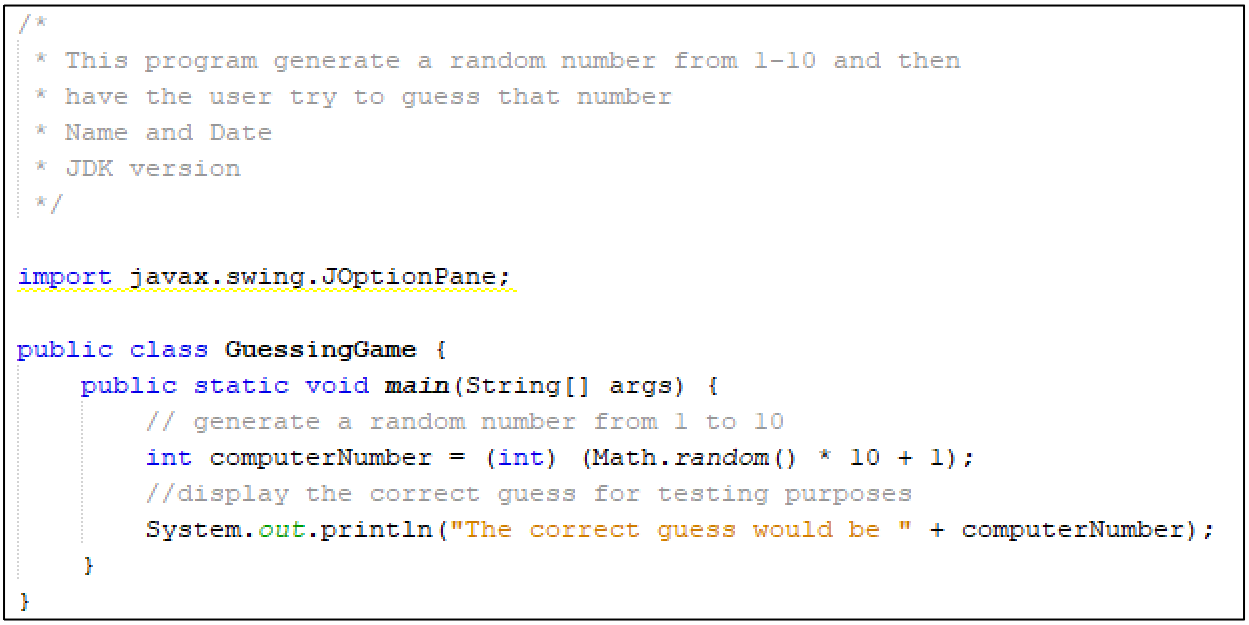
4. However, the game requires the number to be between 1 and 100. Our current statement is going to generate numbers between 0 and 0 because the lowest number of .0000000000000 will be converted to an int of 0 and the highest possible number of .9999999999 will be converted to an int of 0. Remember, that no rounding is done. So let’s multiply by 10 as follows:

   int computerNumber = (int) (Math.random( ) * 10);

5. The above statement will convert the smallest number of .000000 multiplied by 100 to an int of 0 and convert the largest number of .9999999999 multiplied by 100 to an int of 99. BUT, we wanted the numbers to be between 1 and 100 so we need to add 1 to the formula so that our final formula will be:

   int computerNumber = (int) (Math.random( ) * 10 + 1);

6. The above line is the final formula to have the computer generate a number between 1 and 10 and put that random number in the integer variable called computerNumber.
7. Let’s add a line of code to display the randomly generating number for testing purposes. It is a good idea to put print out statements in your code to test whether your program is functioning the way that you think it is.

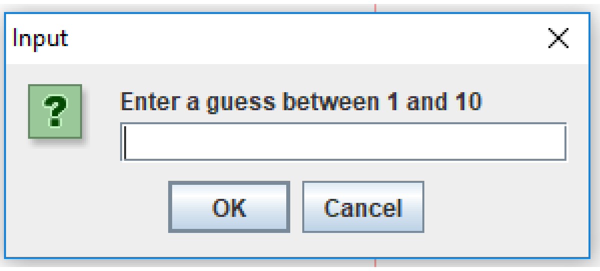


```
/*
 * This program generate a random number from 1-10 and then
 * have the user try to guess that number
 * Name and Date
 * JDK version
 */

import javax.swing.JOptionPane;

public class GuessingGame {
    public static void main(String[] args) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
    }
}
```

8. Next we need to display a window on screen, requesting the user to guess a number between 1 and 10. On this same screen, the user should be able to type in her guess. The statement will be as follows:

   String response = JOptionPane.showInputDialog(null,
   "Enter a guess between 1 and 10");

   The above statement will display a screen as follows:

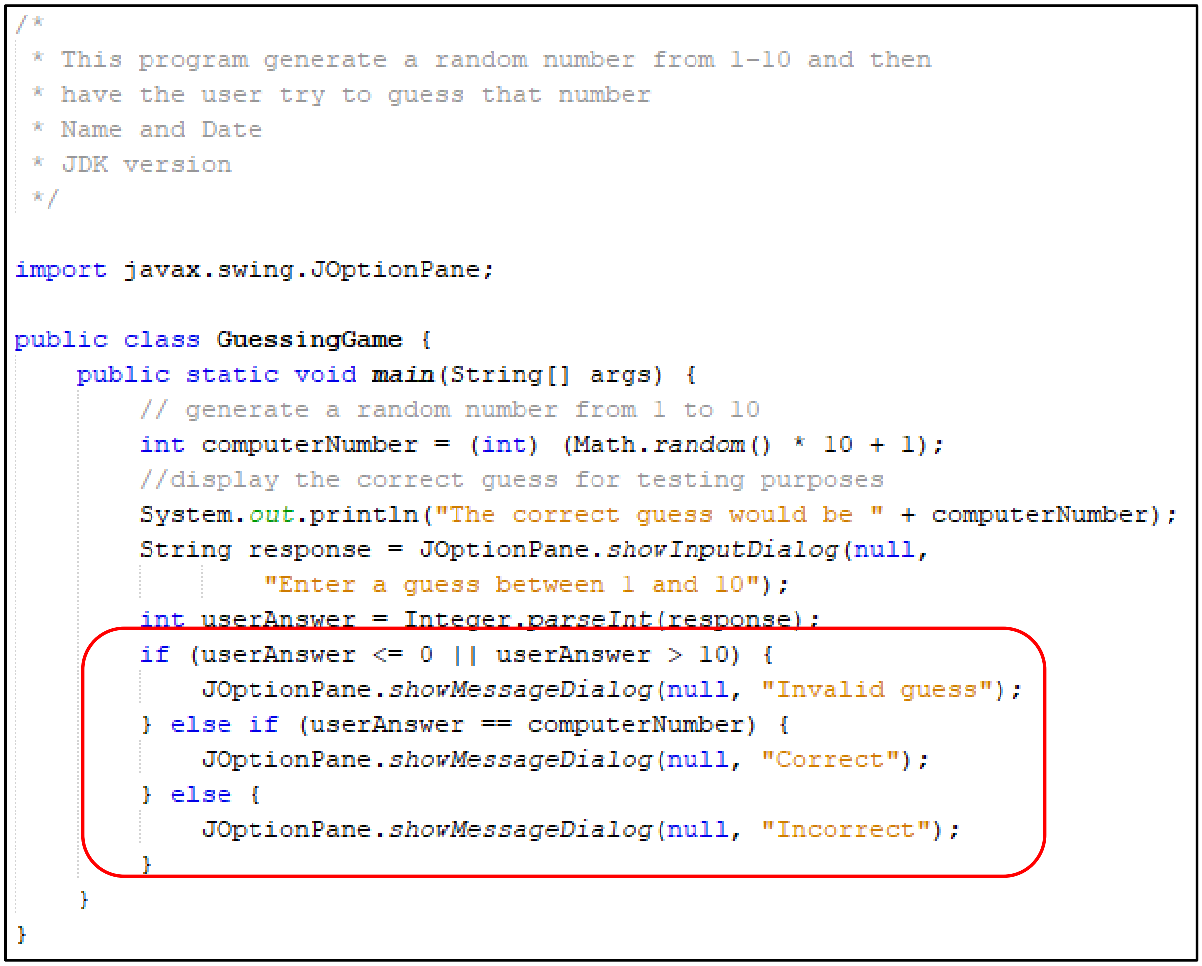

Input

Enter a guess between 1 and 10

9. The user will type in a guess and that guess will be placed into the String variable called response. To use this string as an integer later in the program, it will need to be converted into an int by using the following statement:

   int userAnswer = Integer.parseInt(response);

10. Next, we need to determine whether the guess is invalid, correct, or incorrect. We will add a set of conditionals to do this.



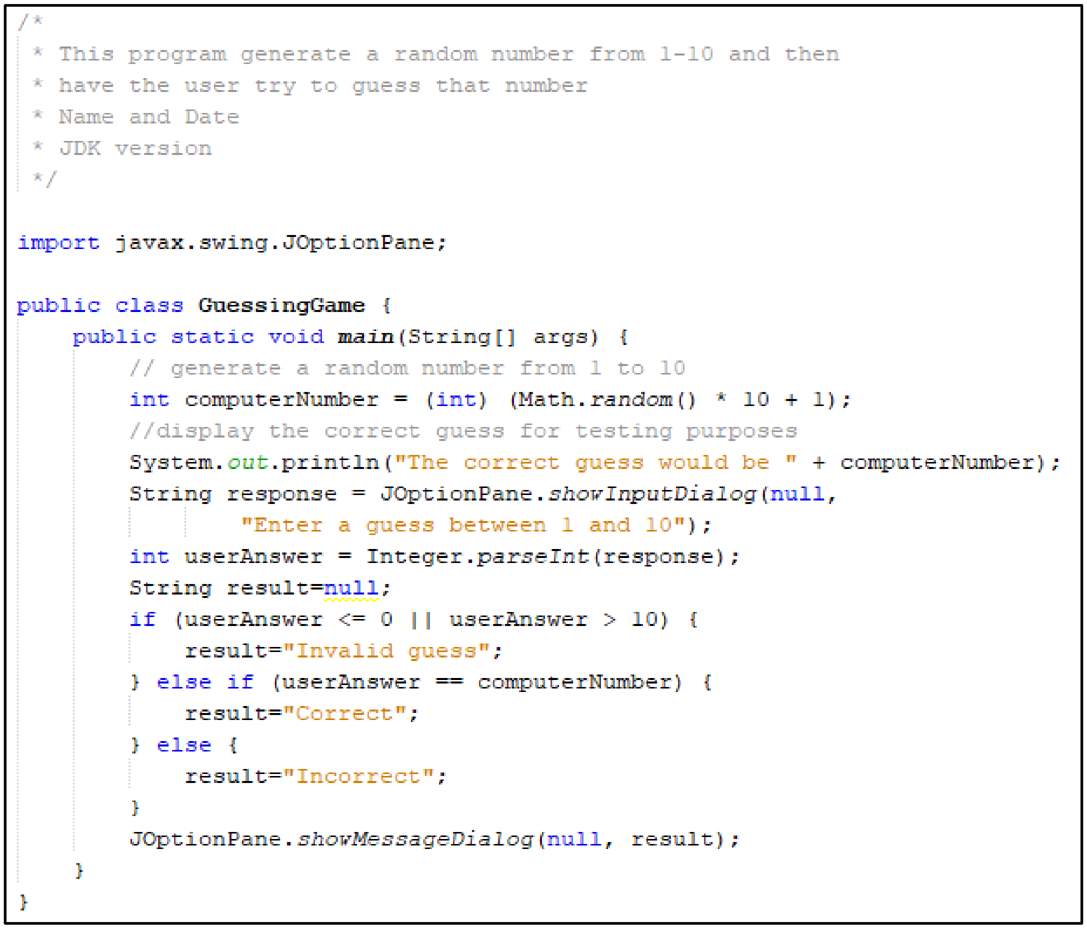
```
/*
 * This program generate a random number from 1-10 and then
 * have the user try to guess that number
 * Name and Date
 * JDK version
 */

import javax.swing.JOptionPane;

public class GuessingGame {
    public static void main(String[] args) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
        String response = JOptionPane.showInputDialog(null,
                "Enter a guess between 1 and 10");
        int userAnswer = Integer.parseInt(response);
        if (userAnswer <= 0 || userAnswer > 10) {
            JOptionPane.showMessageDialog(null, "Invalid guess");
        } else if (userAnswer == computerNumber) {
            JOptionPane.showMessageDialog(null, "Correct");
        } else {
            JOptionPane.showMessageDialog(null, "Incorrect");
        }
    }
}
```

11. To test this program, you need to guess incorrectly once and you need to guess correctly at least once. This way you will see if your *if* statement is working. However, it will be near impossible for you to guess the random number that the computer generated. Thus, we will need some way of seeing the correctNumber before we guess so we can guess correctly (cheating really ☺). The following line after the computer generates its random number helps us to guess correct and incorrect guesses to test our program.

    System.out.println("The correct guess would be" + computerNumber);

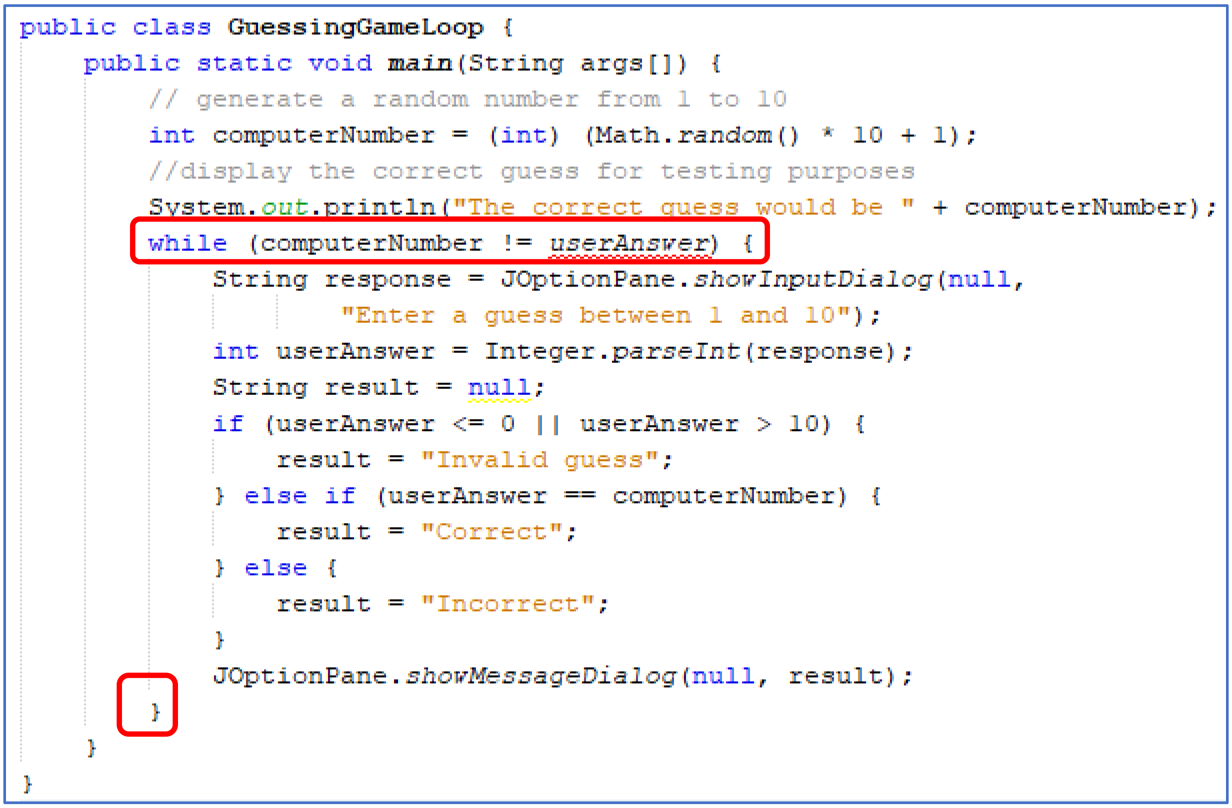
12. Now compile the program and execute it. You should be able to see the guess in the output window at the bottom of NetBeans. Test all the possibilities by guessing 0 (invalid), guessing 125 (invalid), guessing the exact number (correct), and guessing the wrong number (incorrect).

13. You can also use a variable to save the results from the conditionals and print the result after the conditional is execute. The following code will have the same functionality. I choose to put **null** into the **result** variable because I like to initialize my variables as a procaution in case the variable gets printed in the future and never was assigned a value. In this particular example, initializing the variable (giving it a value) is not necessary since the else statement ensures that the **result** variable gets a value.



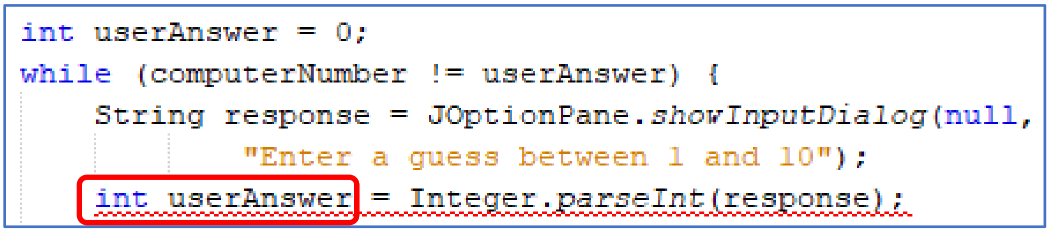
```java
/*
 * This program generate a random number from 1-10 and then
 * have the user try to guess that number
 * Name and Date
 * JDK version
 */

import javax.swing.JOptionPane;

public class GuessingGame {
    public static void main(String[] args) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
        String response = JOptionPane.showInputDialog(null,
                "Enter a guess between 1 and 10");
        int userAnswer = Integer.parseInt(response);
        String result=null;
        if (userAnswer <= 0 || userAnswer > 10) {
            result="Invalid guess";
        } else if (userAnswer == computerNumber) {
            result="Correct";
        } else {
            result="Incorrect";
        }
        JOptionPane.showMessageDialog(null, result);
    }
}
```

14. We are going to modify the guessing game to have a loop so that the user can keep guessing a number until they guess the correct number. Let’s add a **while** loop since we don’t know how many times the user is going to guess the correct number. We could make a for loop work, but normally a **for** loop is better for situations where you know how many times you want the loop to execute. We will compare the computerNumber variable (randomly generated number) with the userAnswer variable (user’s guess). If these 2 variables are not equal (meaning the user did not guess the right number), then the loop will execute. Please add the loop as shown in the following code. You will have an error in your code. Do not run your program. This error will be explained in the next step.



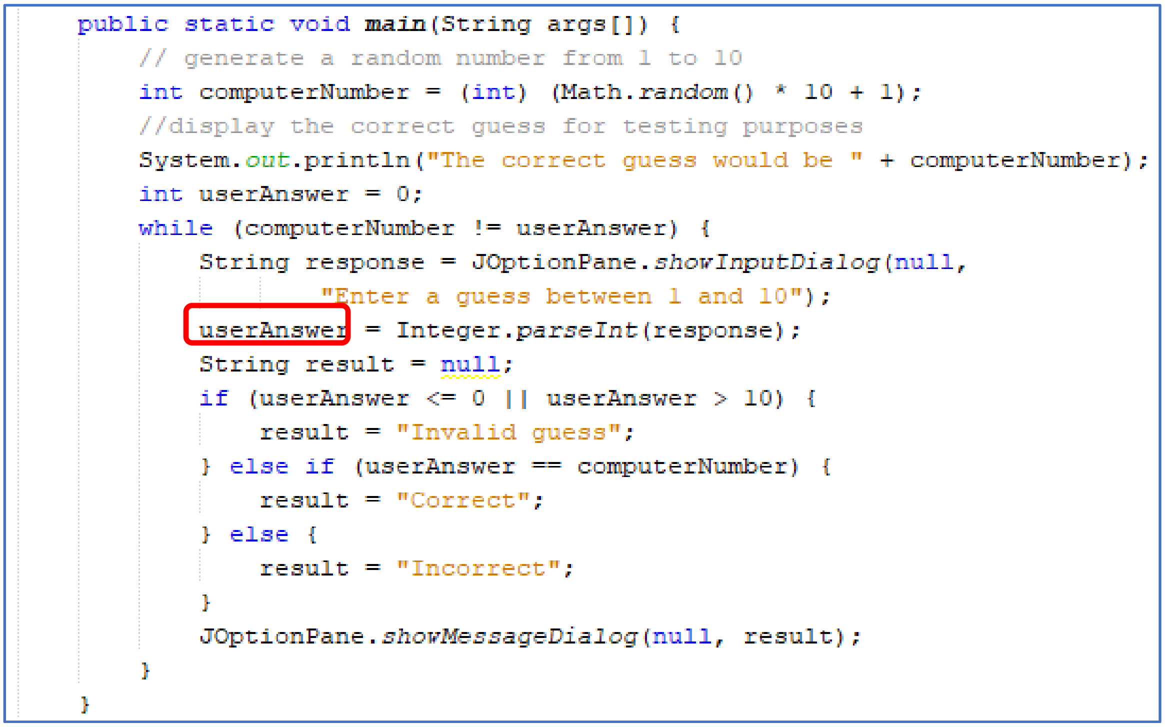
```java
public class GuessingGameLoop {
    public static void main(String args[]) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
        while (computerNumber != userAnswer) {
            String response = JOptionPane.showInputDialog(null,
                    "Enter a guess between 1 and 10");
            int userAnswer = Integer.parseInt(response);
            String result = null;
            if (userAnswer <= 0 || userAnswer > 10) {
                result = "Invalid guess";
            } else if (userAnswer == computerNumber) {
                result = "Correct";
            } else {
                result = "Incorrect";
            }
            JOptionPane.showMessageDialog(null, result);
        }
    }
}
```

15. You should have noticed an error appear in your code. If you hover over the red underline in NetBeans, it should tell you that it cannot find the variable userAnswer. If you locate the declaration of the userAnswer variable, you will notice that it is inside of your loop. We need to move this variable declaration before the loop so that when the while loop condtion is evaluated it knows about this variable. You need to be sure to initialize this variable (give it a value) so that can make the comparison that you are asking it to make. If you don’t give it a value then it won’t be able to determine if it is not equal to the computerNumber variable. Make sure that you initialize this variable with something that will cause your loop to execute the first time. For example, if you were to pick a value of 5 and your program generated a random number of 5 by chance, your loop would not execute since the userAnswer would equal the computerNumber and the loop only executes when they are not equal. Let’s choose 0 as the initialization value for userAnswer. The number of 0 will not cause any issues with our loop criteria since the computerNumber will not be generated as 0 based on our equation. You will notice the following error. We will fix this in the next step.



```
int userAnswer = 0;
while (computerNumber != userAnswer) {
    String response = JOptionPane.showInputDialog(null,
            "Enter a guess between 1 and 10");
    int userAnswer = Integer.parseInt(response);
```

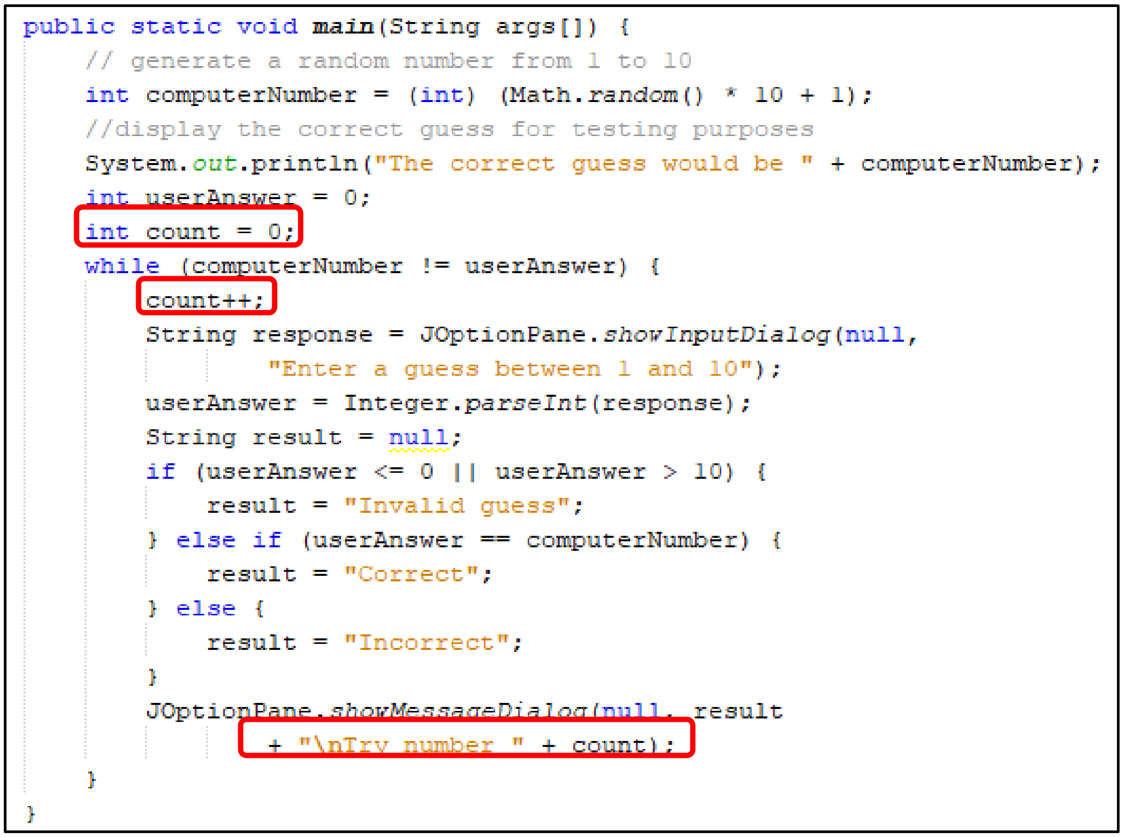
16. We have an error in our code because we declared the userAnswer variable (telling Java that userAnswer was an integer) 2 times. We have to remove the second declaration in front of our userAnswer variable.



```
    public static void main(String args[]) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
        int userAnswer = 0;
        while (computerNumber != userAnswer) {
            String response = JOptionPane.showInputDialog(null,
                    "Enter a guess between 1 and 10");
            userAnswer = Integer.parseInt(response);
            String result = null;
            if (userAnswer <= 0 || userAnswer > 10) {
                result = "Invalid guess";
            } else if (userAnswer == computerNumber) {
                result = "Correct";
            } else {
                result = "Incorrect";
            }
            JOptionPane.showMessageDialog(null, result);
        }
    }
```

17. Run your program and test the new code by guessing a number that is incorrect (remember that you have a print out of the correct number in the output dialog in NetBeans as a cheat to help you with testing your program). Next, enter an invalid number to ensure that it is working correctly. Then guess the correct number. Did it work? Did the program stop running?

18. Next, we will add a counter variable to keep track of how many guesses it takes the user to guess the correct number. Let’s add a variable named count. Then, we will increment this variable by 1 inside our loop. You need to put this code before the output box that will display the number of guesses. Then, add the number of guesses to the output box.

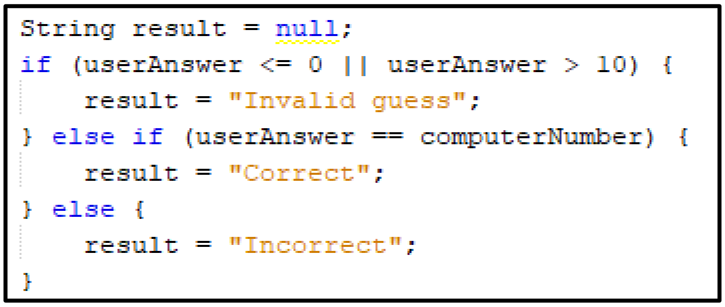


```java
public static void main(String args[]) {
    // generate a random number from 1 to 10
    int computerNumber = (int) (Math.random() * 10 + 1);
    //display the correct guess for testing purposes
    System.out.println("The correct guess would be " + computerNumber);
    int userAnswer = 0;
    int count = 0;
    while (computerNumber != userAnswer) {
        count++;
        String response = JOptionPane.showInputDialog(null,
                "Enter a guess between 1 and 10");
        userAnswer = Integer.parseInt(response);
        String result = null;
        if (userAnswer <= 0 || userAnswer > 10) {
            result = "Invalid guess";
        } else if (userAnswer == computerNumber) {
            result = "Correct";
        } else {
            result = "Incorrect";
        }
        JOptionPane.showMessageDialog(null, result
                + "\nTry number " + count);
    }
}
```

19. Run your program and test the new code by guessing the correct number on the first guess. Did it give the right number of guesses in your output box? Try running your program and guessing the number correct after 3 guesses. Did it work correctly?

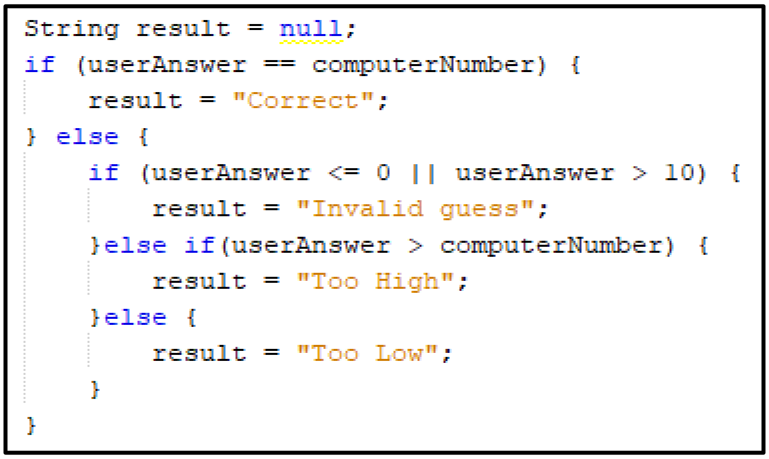
20. Let’s modify our current conditionals to tell the user if their guess is too high, too low, or invalid when they guess an incorrect number. This will help the user to make a better guess. Please adjust the current conditionals to match the modified conditionals.

Current Conditionals:



```java
String result = null;
if (userAnswer <= 0 || userAnswer > 10) {
    result = "Invalid guess";
} else if (userAnswer == computerNumber) {
    result = "Correct";
} else {
    result = "Incorrect";
}
```

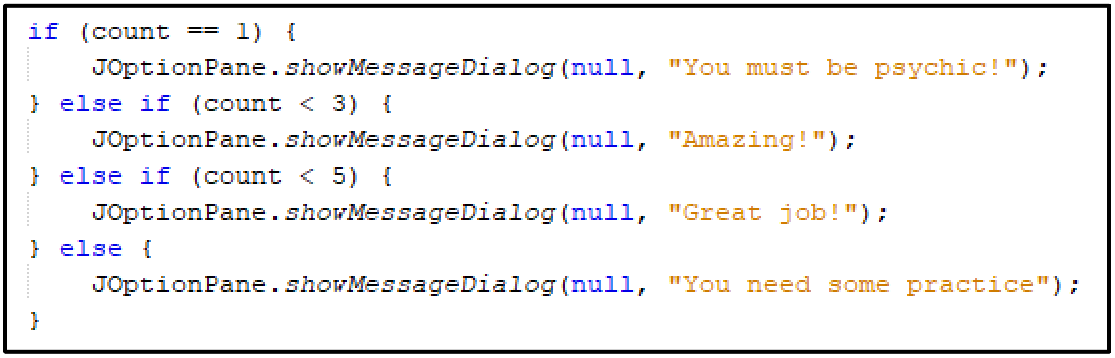
Modified Conditionals:



```
String result = null;
if (userAnswer == computerNumber) {
    result = "Correct";
} else {
    if (userAnswer <= 0 || userAnswer > 10) {
        result = "Invalid guess";
    }else if(userAnswer > computerNumber) {
        result = "Too High";
    }else {
        result = "Too Low";
    }
}
```

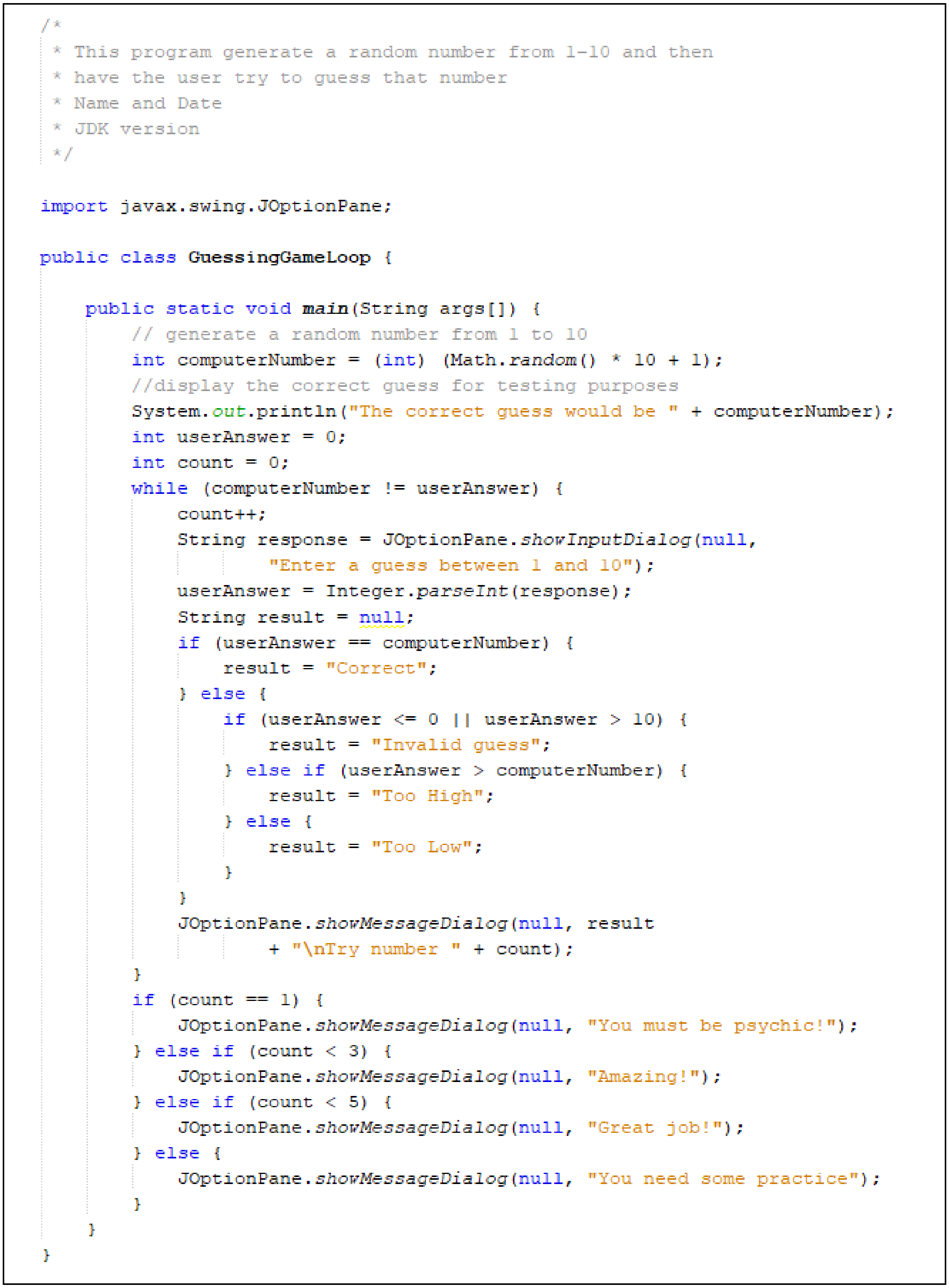
21. Test your program by entering a number that is too high, then test the program by entering a number that it too low. Guess the correct number to stop the game. Did it work correctly?

22. Let’s add messages to the end of the program based on how many guesses it took the user to guess the correct number. This code should go after your while loop. The condition criteria will evaluate the **count** variable to determine which message will be displayed to the user.



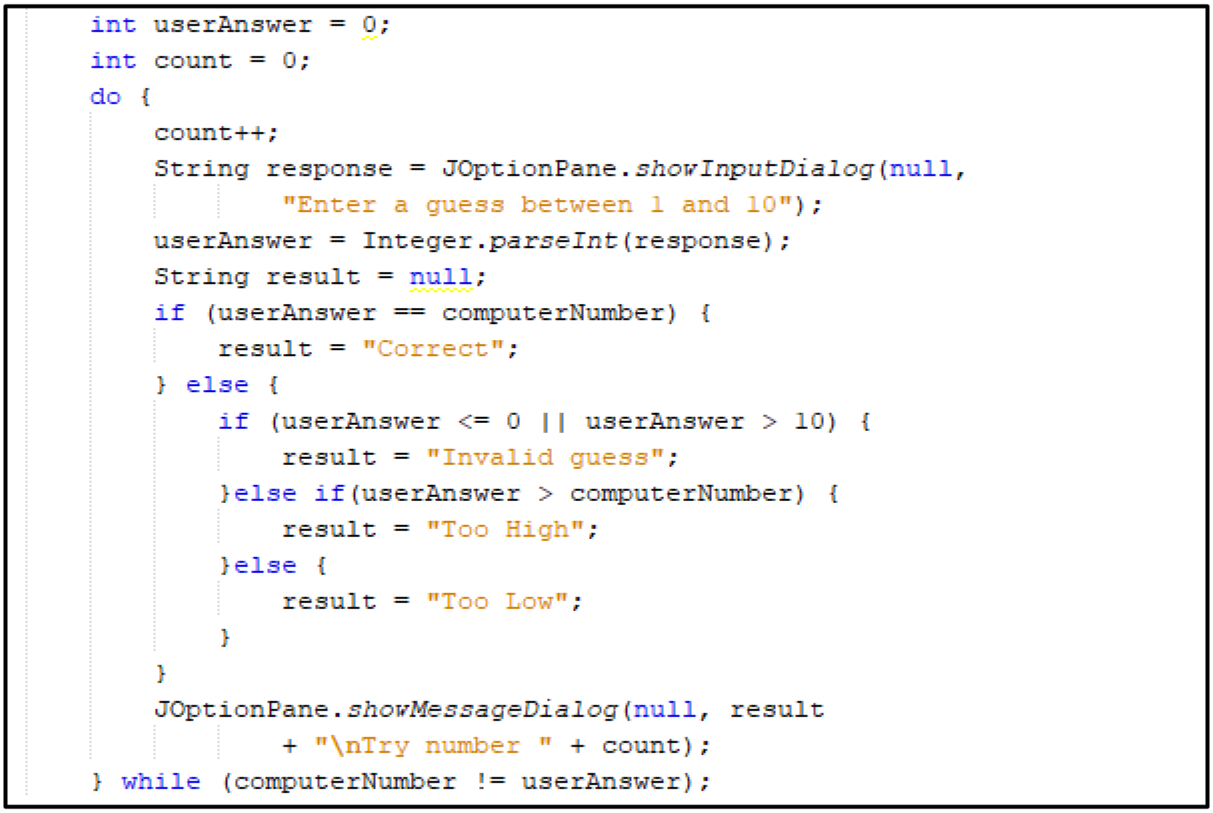
```
if (count == 1) {
    JOptionPane.showMessageDialog(null, "You must be psychic!");
} else if (count < 3) {
    JOptionPane.showMessageDialog(null, "Amazing!");
} else if (count < 5) {
    JOptionPane.showMessageDialog(null, "Great job!");
} else {
    JOptionPane.showMessageDialog(null, "You need some practice");
}
```

23. The final version of the program should look as follows:



```
/*
 * This program generate a random number from 1-10 and then
 * have the user try to guess that number
 * Name and Date
 * JDK version
 */

import javax.swing.JOptionPane;

public class GuessingGameLoop {

    public static void main(String args[]) {
        // generate a random number from 1 to 10
        int computerNumber = (int) (Math.random() * 10 + 1);
        //display the correct guess for testing purposes
        System.out.println("The correct guess would be " + computerNumber);
        int userAnswer = 0;
        int count = 0;
        while (computerNumber != userAnswer) {
            count++;
            String response = JOptionPane.showInputDialog(null,
                    "Enter a guess between 1 and 10");
            userAnswer = Integer.parseInt(response);
            String result = null;
            if (userAnswer == computerNumber) {
                result = "Correct";
            } else {
                if (userAnswer <= 0 || userAnswer > 10) {
                    result = "Invalid guess";
                } else if (userAnswer > computerNumber) {
                    result = "Too High";
                } else {
                    result = "Too Low";
                }
            }
            JOptionPane.showMessageDialog(null, result
                    + "\nTry number " + count);
        }
        if (count == 1) {
            JOptionPane.showMessageDialog(null, "You must be psychic!");
        } else if (count < 3) {
            JOptionPane.showMessageDialog(null, "Amazing!");
        } else if (count < 5) {
            JOptionPane.showMessageDialog(null, "Great job!");
        } else {
            JOptionPane.showMessageDialog(null, "You need some practice");
        }
    }
}
```

24. You can adjust the while loop to be a do/while loop if you wish to do so or you can leave it as a while loop.



```
int userAnswer = 0;
int count = 0;
do {
    count++;
    String response = JOptionPane.showInputDialog(null,
            "Enter a guess between 1 and 10");
    userAnswer = Integer.parseInt(response);
    String result = null;
    if (userAnswer == computerNumber) {
        result = "Correct";
    } else {
        if (userAnswer <= 0 || userAnswer > 10) {
            result = "Invalid guess";
        }else if(userAnswer > computerNumber) {
            result = "Too High";
        }else {
            result = "Too Low";
        }
    }
    JOptionPane.showMessageDialog(null, result
            + "\nTry number " + count);
} while (computerNumber != userAnswer);
```