

A Micro Project report on  
**JAVA NUMBER GUESSING GAME**

Submitted to the CMR Institute of Technology in partial fulfillment of the requirement for the

award of the Laboratory of

**JAVA PROGRAMMING**  
**Of**  
**II B.Tech I Semester**  
**In**  
**Computer Science and Engineering**  
Submitted by

<b>G.MOHAN SAI</b>	<b>(19R01A0514)</b>
<b>K.AKHILESH REDDY</b>	<b>(19R01A0529)</b>
<b>K.ANJAN KUMAR REDDY</b>	<b>(19R01A0589)</b>
<b>G.KOUSHIK</b>	<b>(19R01A0517)</b>

Under the Guidance Of  
**Ms. S. SWETHA**  
(Assistant Professor, CSE Dept)



**CMR INSTITUTE OF TECHNOLOGY**

**(UGC AUTONOMOUS )**

**(Approved by AICTE,Affiliated to JNTU,Kukatpally,Hyderabad)**

**Kandlakoya, Medchal Road, Hyderabad**

**2020-2021**

CMR Institute of Technology

**(UGC AUTONOMOUS)**

**(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad) Kandlakoya, Medchal  
Road, Hyderabad.**



**CERTIFICATE**

**This is to certify that a Micro Project entitled with:**

**' JAVA NUMBER GUESSING GAME'**

**is being**

**Submitted By**

<b>G.MOHAN SAI</b>	<b>(19R01A0514)</b>
<b>K.AKHILESH REDDY</b>	<b>(19R01A0529)</b>
<b>K.ANJAN KUMAR REDDY</b>	<b>(19R01A0589)</b>
<b>G.KOUSHIK</b>	<b>(19R01A0517)</b>

In partial fulfillment of the requirement for the award of the Python Laboratory of II B. Tech I Semester in CSE to the CMRIT, Hyderabad is a record of a bonafide work carried out under our guidance and supervision.

**Signature of Faculty**

**Ms. S Swetha**

(Assistant Professor)

**Signature of HOD**

**Dr. K Pradeep Reddy**

## **ACKNOWLEDGEMENT**

We are extremely grateful to **Dr. M. Janga Reddy, Director, Dr. B. Satyanarayana, Principal,** and **Dr. K. Pradeep Reddy, Head of Department,** Dept of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during the entire duration.

We are extremely thankful to our Python programming faculty in charge **Ms. S Swetha,** Dept of Computer Science and Engineering, CMR Institute of Technology for his constant guidance, encouragement, and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided us directly or indirectly for the successful completion of the project.

<b>G.MOHAN SAI</b>	<b>(19R01A0514)</b>
<b>K.AKHILESH REDDY</b>	<b>(19R01A0529)</b>
<b>K.ANJAN KUMAR REDDY</b>	<b>(19R01A0589)</b>
<b>G.KOUSHIK</b>	<b>(19R01A0517)</b>

# CONTENTS

<b>Particulars</b>	<b>Page No.</b>
1. Introduction	1
2. Algorithm	5
3. Requirements (S/W and H/W)	6
4. Implementation	6
5. Results	8
6. Conclusion	9
7. References	9

## **INTRODUCTION**

The **Guess the number game** can be played easily and need only one player because on the other end there will be a computer playing with you.

Random() method is used to pick a random number. Random\_number.nextInt(100); here 100 denotes that the random number range will be bounded by 100.

int turn is initialized to zero so that it can count the number of turns user has used to guess the right answer. For each iteration, the value of turn will be increased by 1 as we put turn++ in our loop.

The integer “i” is used to count the turns the user has left to guess the number.

## **Java Swing Tutorial**

Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

## **Java JOptionPane**

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.

## **ALGORITHM**

- First, you have to take a random number and store it to an integer type variable. For that I use Random() Method.
- Then get the number from the user.

- Compare that number with the Random number.
- Just use your own logic and limitation of turns and range for the user.
- For score system, you can follow my approach either you can use your own

## **REQUIREMENTS (S/W and H/W)**

### **Minimum System Requirements:**

- Operating Systems and CPU architecture:
- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- RAM and free disk space:
- 4 GB RAM
- 5 GB free disk space

### **Software Requirement:**

- Net Beans IDE
- Web browser with Adobe Acrobat accessibility

## **IMPLEMENTATION**

```
package guessinggame;
import javax.swing.*;

public class Tester {
    public static void main(String[] args) {
        int computerNumber = (int) (Math.random()*100 + 1);
        int userAnswer = 0;
        // System.out.println("The correct guess would be " + computerNumber);
    }
}
```

```

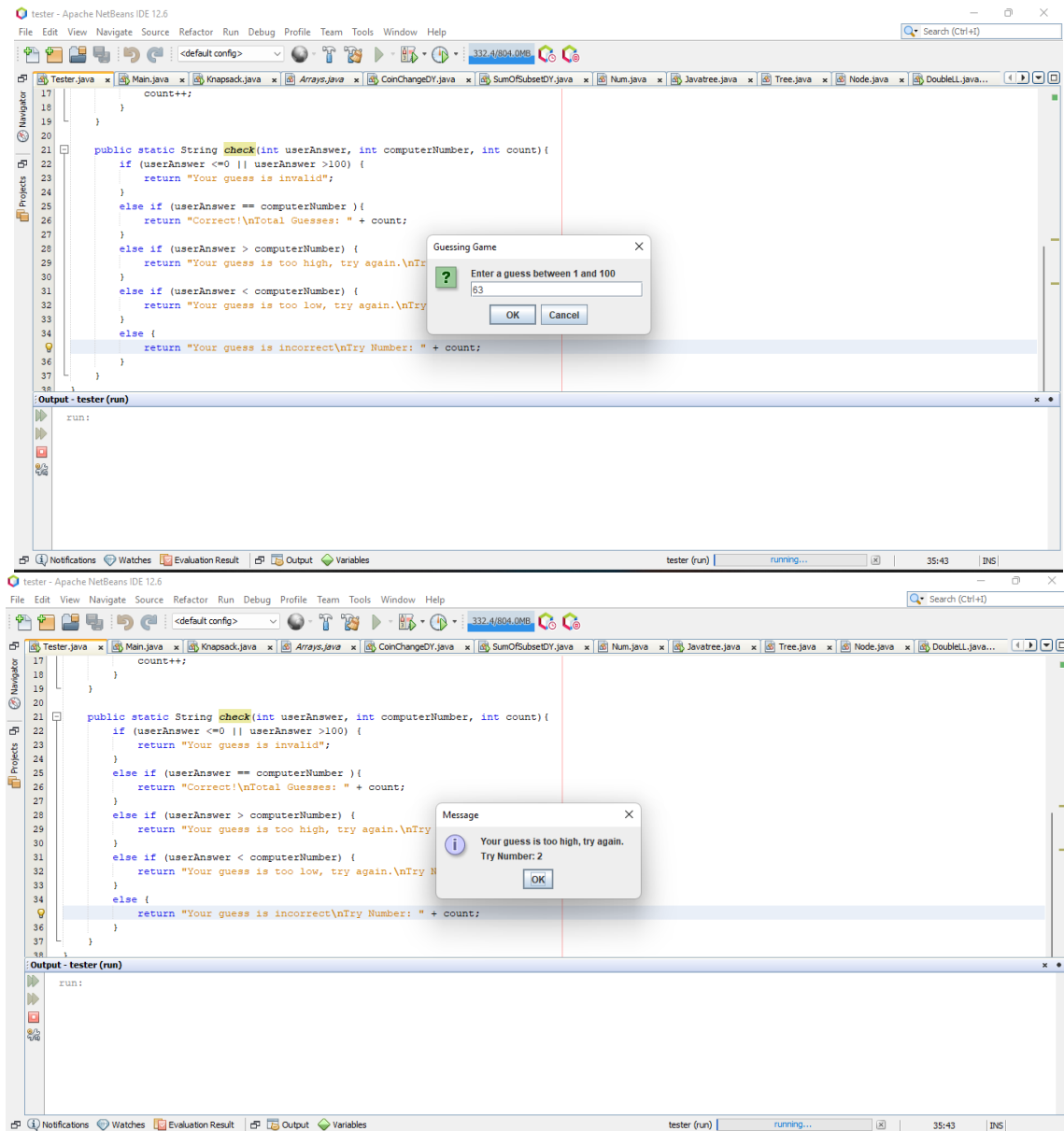
int count = 1;

while (userAnswer != computerNumber)
{
    String response = JOptionPane.showInputDialog(null,
        "Enter a guess between 1 and 100", "Guessing Game", 3);
    userAnswer = Integer.parseInt(response);
    JOptionPane.showMessageDialog(null, ""+ check(userAnswer, computerNumber, count));
    count++;
}
}

public static String check(int userAnswer, int computerNumber, int count){
    if (userAnswer <=0 || userAnswer >100) {
        return "Your guess is invalid";
    }
    else if (userAnswer == computerNumber ){
        return "Correct!\nTotal Guesses: " + count;
    }
    else if (userAnswer > computerNumber) {
        return "Your guess is too high, try again.\nTry Number: " + count;
    }
    else if (userAnswer < computerNumber) {
        return "Your guess is too low, try again.\nTry Number: " + count;
    }
    else {
        return "Your guess is incorrect\nTry Number: " + count;
    }
}
}

```

# RESULTS





## **CONCLUSION**

In general Swing is not thread safe. All Swing components and related classes, unless otherwise documented, must be accessed on the event dispatching thread.

Typical Swing applications do processing in response to an event generated from a user gesture. For example, clicking on a JButton notifies all ActionListeners added to the JButton. As all events generated from a user gesture are dispatched on the event dispatching thread, most developers are not impacted by the restriction.

Where the impact lies, however, is in constructing and showing a Swing application. Calls to an application's main method, or methods in Applet, are not invoked on the event dispatching thread. As such, care must be taken to transfer control to the event dispatching thread when constructing and showing an application or applet. The preferred way to transfer control and begin working with Swing is to use `invokeLater`. The `invokeLater` method schedules a Runnable to be processed on the event dispatching thread. The following two examples work equally well for transferring control and starting up a Swing application:

## **REFERENCES**

- [Java Operators \(w3schools.com\)](http://www.w3schools.com)
- [Java JOptionPane - javatpoint](http://www.javatpoint.com)

