

# DON BOSCO INSTITUTE OF TECHNOLOGY



## Skill Lab: C++ and Java Programming MINI PROJECT REPORT

On

### “Number Guessing Game” 2021-22

**Submitted By:**

Names of students	Roll No.
1. PRATHAM KOTHURVAR	24
2. RAKSHITA KHANTWAL	26
3. RAVEENA PITALE	27
4. RISHABH GIRI	28

**Date of submission:** 10 December 2021

**Under the guidance of:** Ms. Deepali Kayande

**Mini Project Title** : GUESS THE NUMBER

**Institute Name** : Don Bosco Institute of Technology.

**Institute Address** : Premier Automobiles Road, Kurla (West), Mumbai – 400070

**Department** : EXTC

**Class** : SECOND YEAR

**Project Group Members :**

	<b>Names of students</b>	<b>Roll No.</b>
1.	PRATHAM KOTHURVAR	24
2.	RAKSHITA KHANTWAL	26
3.	RAVEENA PITALE	27
4.	RISHABH GIRI	28

**Date of Submission : 10/12/2021**

**Guide : Ms. Deepali Kayande**

## TABLE OF CONTENTS

<b>SR. NO.</b>	<b>CONTENT</b>	<b>PAGE NO.</b>
<b>CHAPTER 1</b>	INTRODUCTION	4
<b>CHAPTER 2</b>	PROBLEM DEFINITION MODULES (IF ANY)	5
<b>CHAPTER 3</b>	ALGORITHM	5
<b>CHAPTER 4</b>	IMPLEMENTATION	6
<b>CHAPTER 5</b>	RESULTS(SNAPSHOTS)	12
<b>CHAPTER 6</b>	CONCLUSION	14
<b>CHAPTER 7</b>	REFERENCES	15

# CHAPTER 1

## INTRODUCTION

**The task is to write a Java program in which a user will have to guess a randomly generated number.**

**Below are the rules of the game:**

1. The approach is to generate a random number using Math.random() method in Java.
2. Now using if-else loop, take the input from the user and for each input print whether the number is smaller or larger than the actual number which was generated.
3. If the user guessed the number correctly, print that the user won.
4. The user can play this game n number of times.
5. The best score of the user is stored in the memory.
6. If, the player wants to quit, the player can press the give up button. The randomly generated number will then pop up.
7. At the end of the game, the number is revealed along with the number of guesses it took to get the correct number.

## **CHAPTER 2**

### **PROBLEM DEFINITION**

Here, we r trying to make a guessing game. The challenge is to write a Java program in which user will get a randomly generated number. The user then has to guess the randomly generated number. If the user guesses the number in the least number of trials, the number will be stored as the Best score. The user can play the game many number of times so, after completing the game, the user should be able to play again. When the user guesses the number correctly, print that the user won.

## **CHAPTER 3**

### **ALGORITHM**

#### **Algorithm:**

1. Start
2. Create a class Guess
3. Create a method called as `Math.random()*100`. This will generate any random number from 1-100.
4. Create a private class.
5. In it, create an if-else loop.
6. If the users guess is less than the random number, print the number is too low, try again.
7. If the users guess is greater than the random number, print the number is too high, try again.
8. If the user guesses the correct number, print correct, you win!
9. If the user does not want to play the game and clicks on quit button, print the random number generated.
10. Store the least number of tries a user takes to play the game as best score
11. In main function, make an object of class guess.
12. Stop

## CHAPTER 4

# IMPLEMENTATION

//MP Number Guessing Game

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class Guess extends JFrame
{
    private static final long serialVersionUID = 1L;
    JTextField t1, t2, t3, t4;
    JLabel j4;
    ButtonListener bl1;
    ButtonListener2 bl2;
    ButtonListener3 bl3;

    //setting random number in rand variable
    int rand=(int) (Math.random()*100);
    int count=0;

    public Guess()
    {
        //Get the container
        Container c = getContentPane();

        //Set absolute layout
        c.setLayout(null);

        //Set Background Color
        c.setBackground(Color.WHITE);
```

```
//Creating image
JLabel lblpic = new JLabel("");
lblpic.setIcon(new ImageIcon("images.png"));
lblpic.setBounds(0,0,500,350);

//Creating label Guess my number text
JLabel j=new JLabel("Guess my number game");
j.setForeground(Color.RED);
j.setFont(new Font("tunga",Font.BOLD,24));
j.setSize(290,30);
j.setLocation(180,50);

//Creating label Enter a number
JLabel j1=new JLabel("Enter a number between 1-100");
j1.setFont(new Font("tunga",Font.PLAIN,17));
j1.setSize(250,20);
j1.setLocation(200,90);

//Creating TextField for input guess
t1=new JTextField(10);
t1.setSize(50,30);
t1.setLocation(270,120);

//Creating Label for Display message
j4=new JLabel("Try and guess my number");
j4.setFont(new Font("tunga",Font.PLAIN,17));
j4.setSize(200,20);
j4.setLocation(200,160);

//Creating Text field for best score
t2=new JTextField(10);
t2.setSize(40,20);
t2.setLocation(10,10);
```

```
//Creating Label for best score
JLabel j5=new JLabel("Best Score");
j5.setFont(new Font("tunga",Font.PLAIN,17));
j5.setSize(270,20);
j5.setLocation(60,10);
```

```
//Creating guess text field
t3=new JTextField(10);
t3.setSize(40,20);
t3.setLocation(160,10);
```

```
//Creating guess Label
JLabel j6=new JLabel("Guesses");
j6.setFont(new Font("tunga",Font.PLAIN,17));
j6.setSize(270,20);
j6.setLocation(210,10);
```

```
//creating 3 buttons
JButton b1=new JButton("Guess");
b1.setSize(150,40);
b1.setLocation(220,250);
b1=new ButtonListener();
b1.addActionListener(b1);
```

```
JButton b2=new JButton("Give up!");
b2.setSize(100,30);
b2.setLocation(160,200);
b2=new ButtonListener2();
b2.addActionListener(b2);
```

```
JButton b3=new JButton("Play Again");
```



```

b3.setSize(120,30);
b3.setLocation(320,200);
bl3=new ButtonListener3();
b3.addActionListener(bl3);

//Place the components in the pane
c.add(j5);
c.add(j4);
c.add(lblpic);
c.add(j);
c.add(j1);
c.add(t1);
c.add(t2);
c.add(t3);
c.add(b1);
c.add(b2);
c.add(b3);
c.add(j6);

//Changing TextFields to UnEditable
t2.setEditable(false);
t3.setEditable(false);

//Set the title of the window
setTitle("GUESS MY NUMBER");

//Set the size of the window and display it
setSize(650,400);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}

```

```
private class ButtonListener implements ActionListener
```

```

{
    int bestScore=100;
    public void actionPerformed(ActionEvent e)
    {
        int a = Integer.parseInt(t1.getText());
        count=count+1;
        if(a<rand)
        {
            j4.setText(a+" is too low, try again!!");
        }
        else if(a>rand)
        {
            j4.setText(a+" is too high, try again!!");
        }
        else
        {
            j4.setText("CORRECT, YOU WIN!!!!");
            if(count<bestScore)
            {
                bestScore=count;
                t2.setText(bestScore+"");
            }
            else
            {
                t2.setText(""+bestScore);
                t1.setEditable(false);
            }

            //setting focus to input guess text field
            t1.requestFocus();
            t1.selectAll();

            t3.setText(count+"");
        }
    }
}

```

```
}
```

```
private class ButtonListener2 implements ActionListener
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        t3.setText("");
```

```
        j4.setText(rand+" is the answer!");
```

```
        t1.setText("");
```

```
        t1.setEditable(false);
```

```
    }
```

```
}
```

```
private class ButtonListener3 implements ActionListener
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        rand=(int) (Math.random()*100);
```

```
        t1.setText("");
```

```
        t3.setText("");
```

```
        j4.setText("Try and guess my number");
```

```
        t3.setText("");
```

```
        count=0;
```

```
        t1.setEditable(true);
```

```
        t1.requestFocus();
```

```
    }
```

```
}
```

```
public static void main(String[] args)
```

```
{
```

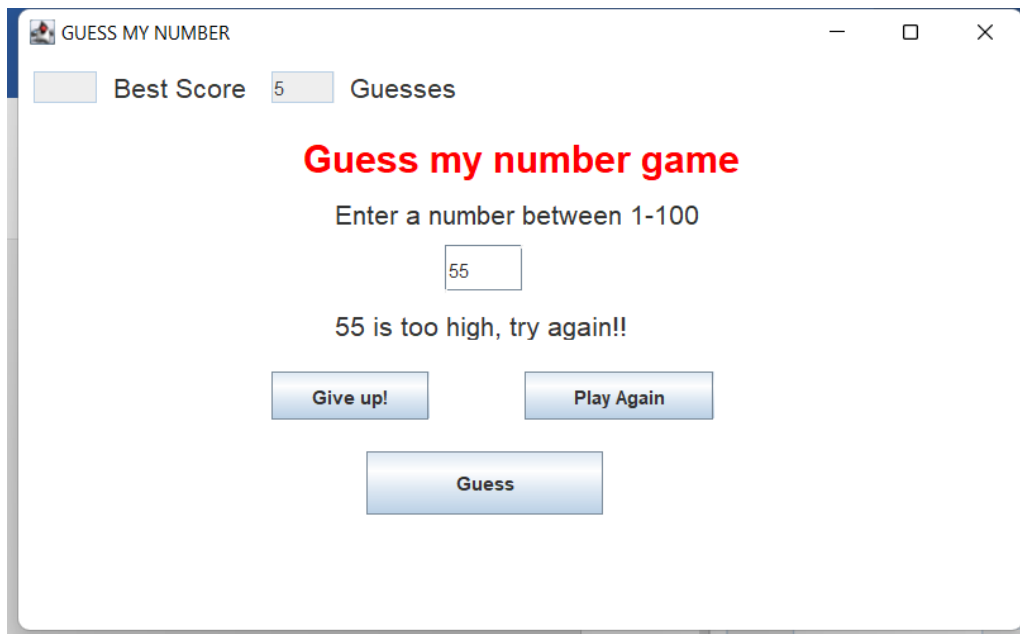
```
    new Guess();
```

```
}
```

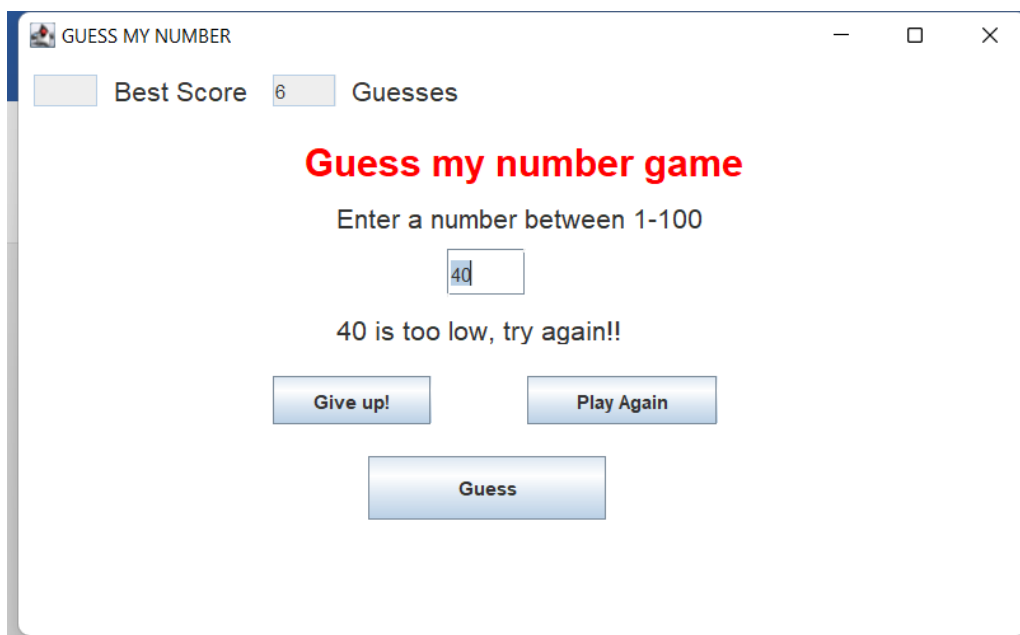
```
}
```

## CHAPTER 5

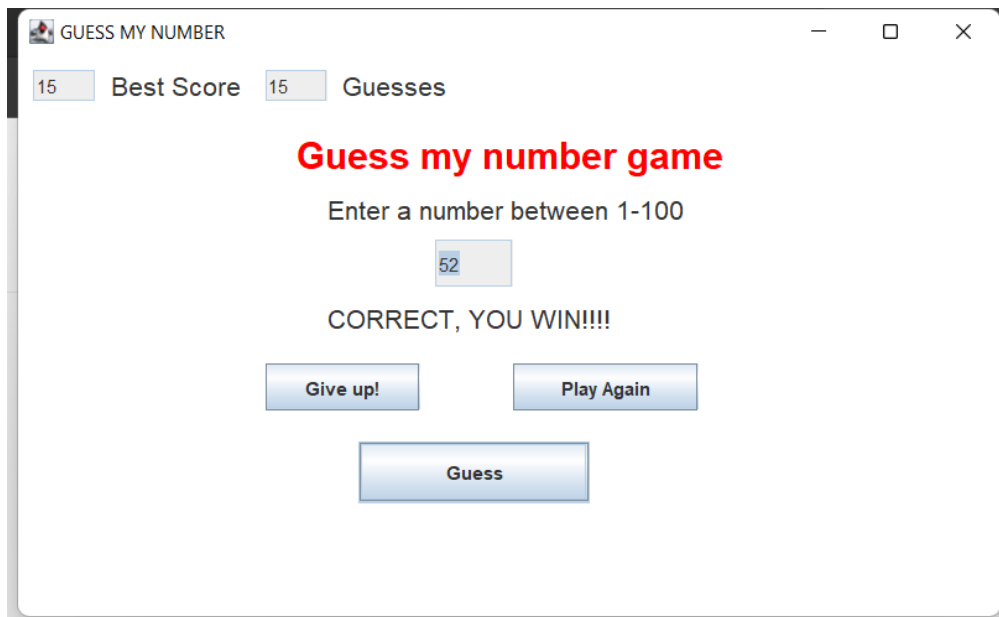
### RESULTS (SNAPSHOTS)



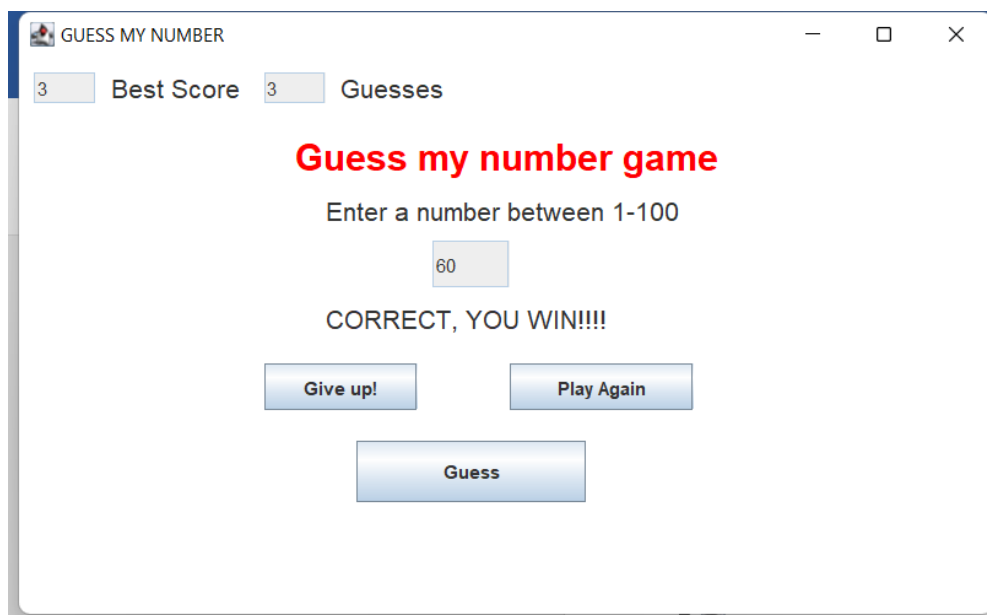
1. When the number entered is higher than the random number.



2. When the number entered is lower than the random number.



3. When the number entered is equal to the random number.



4. When the user plays again and the number is guessed in lesser trials, the best score changes.

## **CHAPTER 6**

### **CONCLUSION**

- Our guessing game is where you put a number as an input, any number of your choice.
- The number will check the conditions, if it satisfy the condition, the number you guessed as input is correct.
- If your guessed number is incorrect, it'll ask you to try again, it'll keep happening until you guess it right. You also have the option to give up. If you choose the give up option, the correct number will be displayed.
- This game also maintains high score, the lesser tries you take the more points you get which keeps you at the top on leaderboard.

## CHAPTER 7

### REFERENCES

- Schildt's Classic java reference updated and expanded for java S.E -6.
- Beginning Java Server Pages[Chopra, Vivek/ Others Wiley- Dreamtech India].
- Programming With Java[Balagurusamy, E. 2nd ed TMH].
- Glenn McCluskey's excellent Java newsletters.
- Youtube references:- <https://www.youtube.com/watch?v=UxzUgVKyQM>  
<https://www.youtube.com/watch?v=-0FXr8P9NNY>
- The Java Tutorial (<https://docs.oracle.com/javase/tutorial/>)
- *Geeks for Geeks*(<https://www.geeksforgeeks.org/java/>)