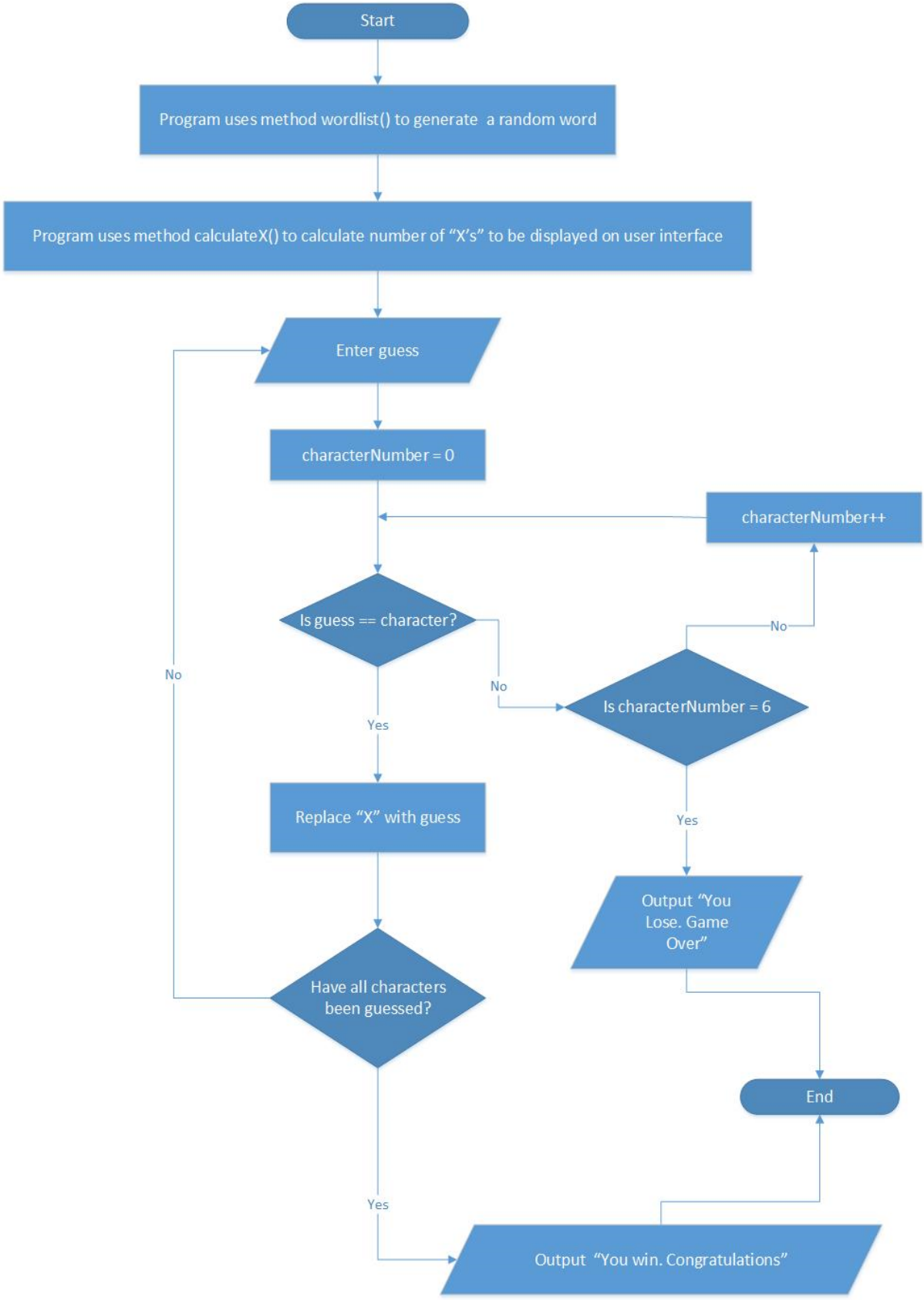
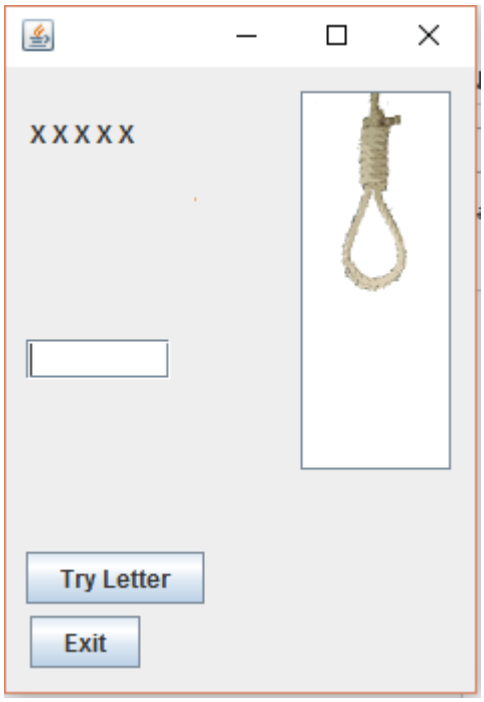
Mewantha Manujaya Bandara, UoW ID: w1583056, IIT ID: 2015045

Flow chart for the program:

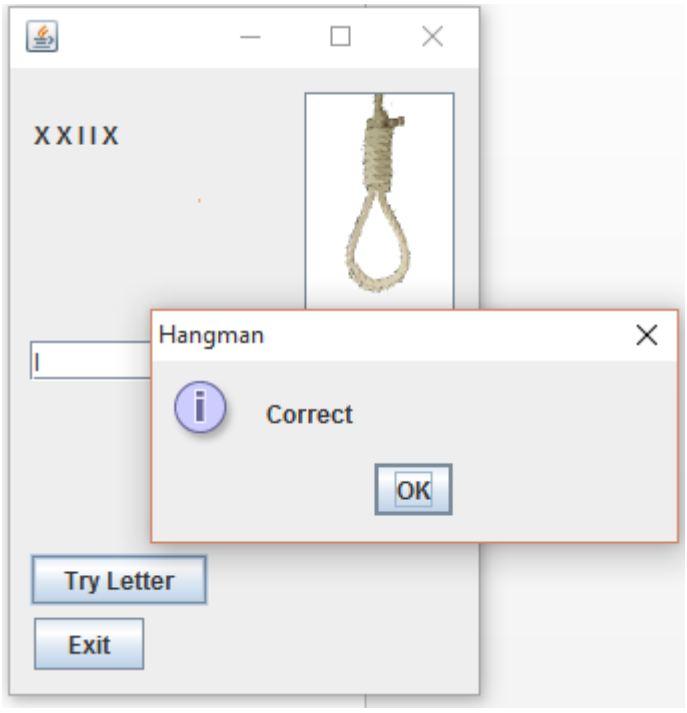


User Interface

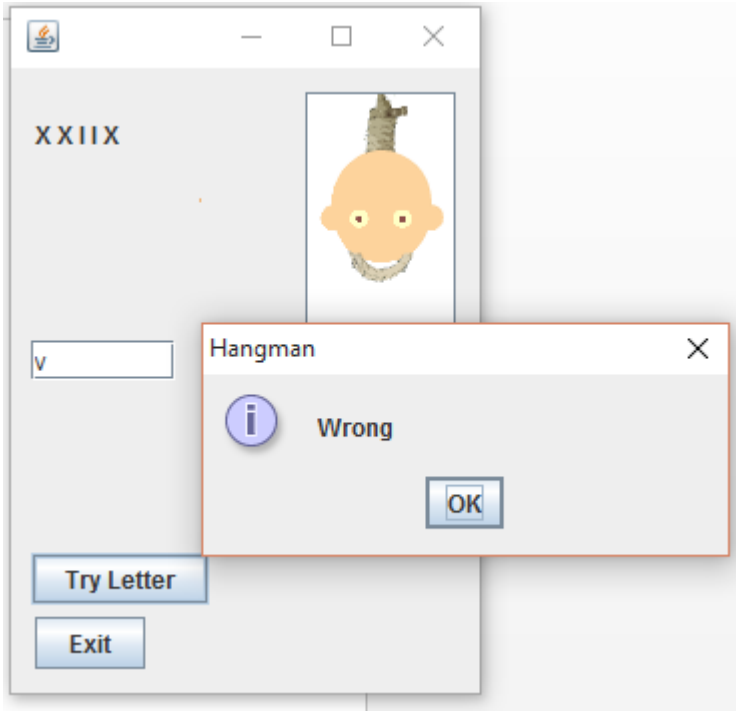
Initial interface:



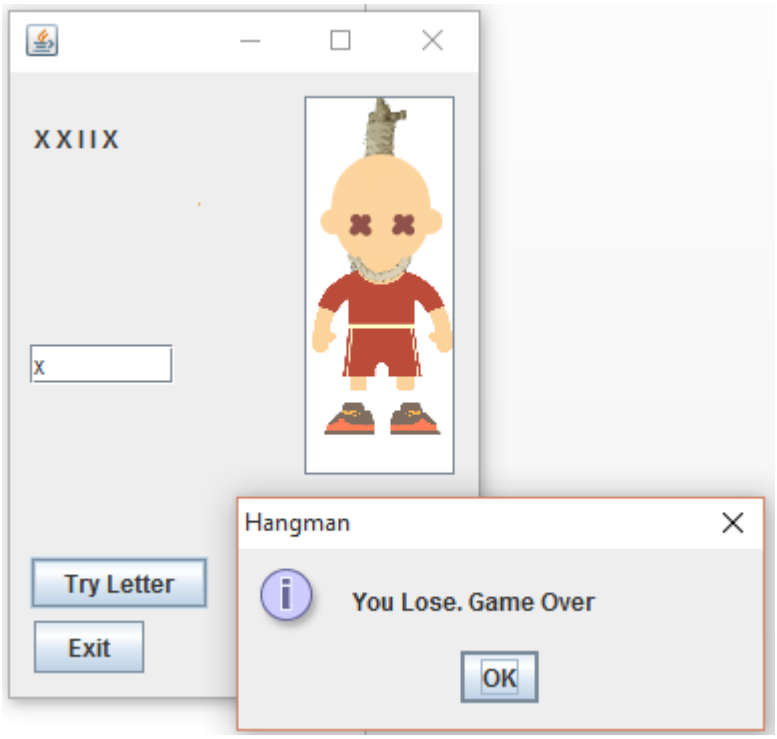
When correct letter has been input:



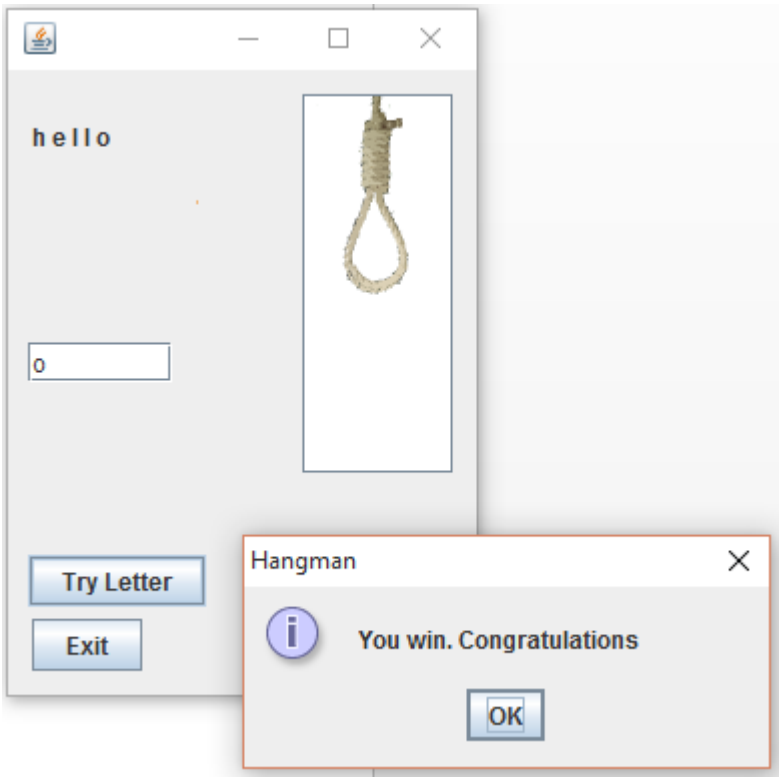
When wrong letter has been input



When number of tries have exceeded the allocated amount:



When game has been won:



Main program body:

```
private void btnTryActionPerformed(java.awt.event.ActionEvent
evt) {
        body(); //Calls main body method
    }

    private void body(){  //Main body method

         char[] wordArray = currentWord;
        char wordGuess;
        StringBuilder guess = new
StringBuilder(wordHolder.getText());

        wordGuess = entryField.getText().charAt(0);
        wordGuess = Character.toLowerCase(wordGuess);

        boolean flag = false;
        int index = 0;
        int indexc = 0;
        boolean flag2 = false;
        while (index < (wordArray.length)) {
            if (wordGuess == (wordArray[index])) {
                flag = true;
                indexc = index;
                guess.setCharAt(2 * index, wordGuess);
                String stringguess = guess.toString();
                wordHolder.setText(stringguess);

            } else {
                flag = false;

            }
            index++;
            if (flag == true) {
                flag2 = true;
            }

        }

        if (flag2 == true) {
            JOptionPane.showMessageDialog(null, "Correct",
"Hangman", JOptionPane.INFORMATION_MESSAGE);
            numberCorrect++;

            if (numberCorrect == wordArray.length) {
                JOptionPane.showMessageDialog(null, "You win.
Congratulations", "Hangman", JOptionPane.INFORMATION_MESSAGE);
                System.exit(0);
            }
        } else {
```

```
            JOptionPane.showMessageDialog(null, "Wrong",
"Hangman", JOptionPane.INFORMATION_MESSAGE);
            numberOfTries++;
            if (numberOfTries == 1) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h1.p
ng")));
            } else if (numberOfTries == 2) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h2.p
ng")));
            } else if (numberOfTries == 3) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h3.p
ng")));
            } else if (numberOfTries == 4) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h4.p
ng")));
            } else if (numberOfTries == 5) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h5.p
ng")));
            } else if (numberOfTries == 6) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h6.p
ng")));
                JOptionPane.showMessageDialog(null, "You Lose.
Game Over", "Hangman", JOptionPane.INFORMATION_MESSAGE);
                System.exit(0);
            }
        }
    }
```

Variables used in user interface :

```
private javax.swing.JButton btnExit; //Exit button
private javax.swing.JButton btnTry;  //Try Letter button
private javax.swing.JTextField entryField; //Enter guess
private javax.swing.JButton hangStatus; //Number of tries
private javax.swing.JLabel wordHolder; //Holds the X's
```

Variables declared in main class :

```
public char[] currentWord; //Array to hold word to be guessed
public int numberOfTries; //Times wrong answer entered
public int numberCorrect; //Times correct answer entered
```

Code extras: Write 6 different hidden words into your program and make it chose a random word each time.

```
public Hangman() {
        initComponents();
        Random ran = new Random();
        int num = ran.nextInt(7 - 1) + 1;
        currentWord = wordlist(num); //Choose random word
        int arrLength = currentWord.length;
        calculateX(arrLength);

    }

    public final char[] wordlist(int num) {//List with words
        char[] v = {'h', 'e', 'l', 'l', 'o'};
        char[] w = {'t', 'r', 'a', 'm', 'p', 'o', 'l', 'i',
'n', 'e'};
        char[] z = {'d', 'e', 's', 'i', 'g', 'n'};
        char[] x = {'h', 'i', 's', 't', 'o', 'r', 'y'};
        char[] c = {'f', 'o', 'r', 'm'};
        char[] b = {'c', 'e', 'n', 't', 'i', 'p', 'e', 'd',
'e'};

        char[] arr = null;
        switch (num) {
            case 1:
                arr = v;
                break;
            case 2:
                arr = w;
                break;
            case 3:
                arr = z;
                break;
            case 4:
                arr = x;
                break;
            case 5:
                arr = c;
            case 6:
                arr = b;

        }
        return arr;

    }
```

Code extras: Make the program work for any length hidden word from 3 to 10 characters, and only display sufficient 'X's to match the length of the hidden word.

```
public Hangman() {
        initComponents();
        Random ran = new Random();
        int num = ran.nextInt(7 - 1) + 1;
        currentWord = wordlist(1);
        int arrLength = currentWord.length;
        calculateX(arrLength); //Calls method to calculate X's


    }
private void calculateX(int length) {

        switch (length) {
            case 3:
                wordHolder.setText("X X X");
                break;
            case 4:
                wordHolder.setText("X X X X");
                break;
            case 5:
                wordHolder.setText("X X X X X");
                break;
            case 6:
                wordHolder.setText("X X X X X X");
                break;
            case 7:
                wordHolder.setText("X X X X X X X");
                break;
            case 8:
                wordHolder.setText("X X X X X X X X");
                break;
            case 9:
                wordHolder.setText("X X X X X X X X X");
                break;
            case 10:
                wordHolder.setText("X X X X X X X X X X");
                break;
        }
    }
```

Code extras: Work out a way to make the program display an appropriate 'Game Over' message.

```
if (flag2 == true) {
            JOptionPane.showMessageDialog(null, "Correct",
"Hangman", JOptionPane.INFORMATION_MESSAGE);
            numberCorrect++;

            if (numberCorrect == wordArray.length) {
                JOptionPane.showMessageDialog(null, "You win.
Congratulations", "Hangman", JOptionPane.INFORMATION_MESSAGE);
                System.exit(0); //If game is won

            }
        } else {
            JOptionPane.showMessageDialog(null, "Wrong",
"Hangman", JOptionPane.INFORMATION_MESSAGE);
            numberOfTries++;
            if (numberOfTries == 1) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h1.p
ng")));
            } else if (numberOfTries == 2) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h2.p
ng")));
            } else if (numberOfTries == 3) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h3.p
ng")));
            } else if (numberOfTries == 4) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h4.p
ng")));
            } else if (numberOfTries == 5) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h5.p
ng")));
            } else if (numberOfTries == 6) {
                hangStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/coursework/h6.p
ng")));
                JOptionPane.showMessageDialog(null, "You Lose.
Game Over", "Hangman", JOptionPane.INFORMATION_MESSAGE);
 //If game is lost

                System.exit(0);
            }
        }
```