# Front PageTable of Contents

[Chapter 1: Definition of the Problem 3](#_Toc47773107)

[Description of the scope of the problem to be tackled 3](#_Toc47773108)

[Statement of the results required 3](#_Toc47773109)

[Details of the input information required 3](#_Toc47773110)

[Chapter 2: Solution of the Problem 5](#_Toc47773111)

[Algorithm Design 5](#_Toc47773112)

[Computer listing of the program 5](#_Toc47773113)

[Details of any special design features 5](#_Toc47773114)

[Chapter 3: Running the Program 6](#_Toc47773115)

[Evidence that the solution works 6](#_Toc47773116)

[Plan of Test Data 7](#_Toc47773117)

[Chapter 4: User Instruction 8](#_Toc47773118)

[Loading the Program 8](#_Toc47773119)

[User Manual 8](#_Toc47773120)

[Chapter 5: Comments and Conclusions 8](#_Toc47773121)

[Limitations and Improvement 8](#_Toc47773122)

# Chapter 1: Definition of the Problem

## Description of the scope of the problem to be tackled

* Description of your project.
* Why did you choose this project?
* Why did you use JAVA programming language?
* What are the advantages of having your project computerised?

## Statement of the results required

* List and explain the elements that your software displays during runtime, such as: main menu, questions and multiple-choice answers, details of how to use the software etc…
* List and describe the variables / constants / arrays used for output. *A sample is shown below:*

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Type** | **Description** |
| **studentList** | Array of String | To hold on the names of the students inputted by the user and displays them on screen. |
| **total** | int | To store the answer of the addition of two numbers and then displays the result on screen. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Details of the input information required

* List and explain the data that the user inputs during runtime, such as: option in main menu, answer to the questions, students’ names and their exam mark, etc.
* List and describe the variables / constants / arrays used for input. *A sample is shown below:*

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Type** | **Description** |
| **Choice** | int | To hold the user’s choice inputted for the main menu. |
| **userAns** | char | Holds the answer replied by the user while playing the quiz (a, b or c). |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Chapter 2: Solution of the Problem

## Algorithm Design

1. List and describe the classes used.
2. List and describe the methods of each class, for example:

* **Hangman** class has seven methods which are
* mainMenu() : It displays the menu and allows the user to enter an option. See flowchart 1.
* play(): This methods initialises the array with questions and answers. Moreover, it displays the questions and the multiple-choice answers, and allows the user to choose an answer. For every correct answer the score is increased by 1, otherwise lives are decreased. See flowchart 2
* Etc….
* **mainHangMan** class contains only one method which is the main method. This method includes an object of class **Hangman** which is used to run the program.
* **Keyboard** class is used as an external class that provides the user the ability to input from the keyboard by using the method readInt(), readChar(), etc…

1. Flowcharts of any complex routines, such as: Main Menu, Play Quiz, Input Students/Marks, etc.

## 

## Computer listing of the program

Ideally use Courier New font, size 10 and line spacing: 1.5pts

## Details of any special design features

Make a list of any special features that you included in your project, such as:

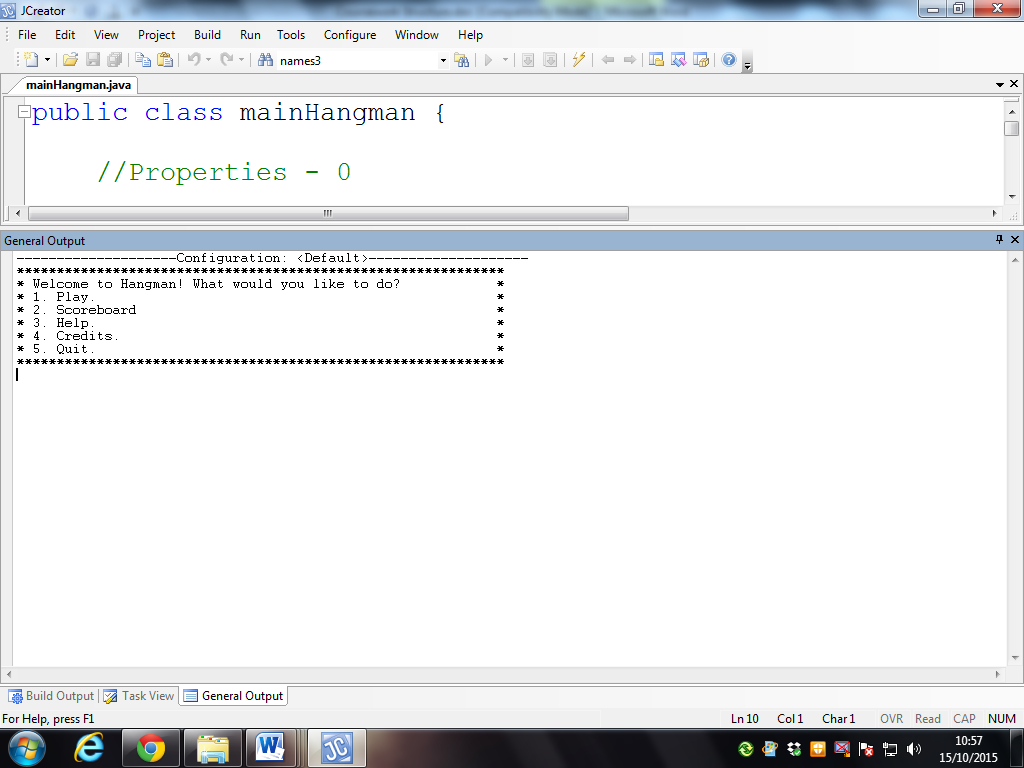
* Graphical Interface (Joption Pane),
* Realisim, for example including randomization of the questions shown (if project is a quiz) or displaying the actual date and time (using beyond o-level code),
* Nice interface for the user by including frames along menus or projection of a logo,
* Auto calculations of average marks, or highest and lowest marks etc.,
* Research and appropriately use of code snippets,
* The proper use of Objects,
* Etc.

# Chapter 3: Running the Program

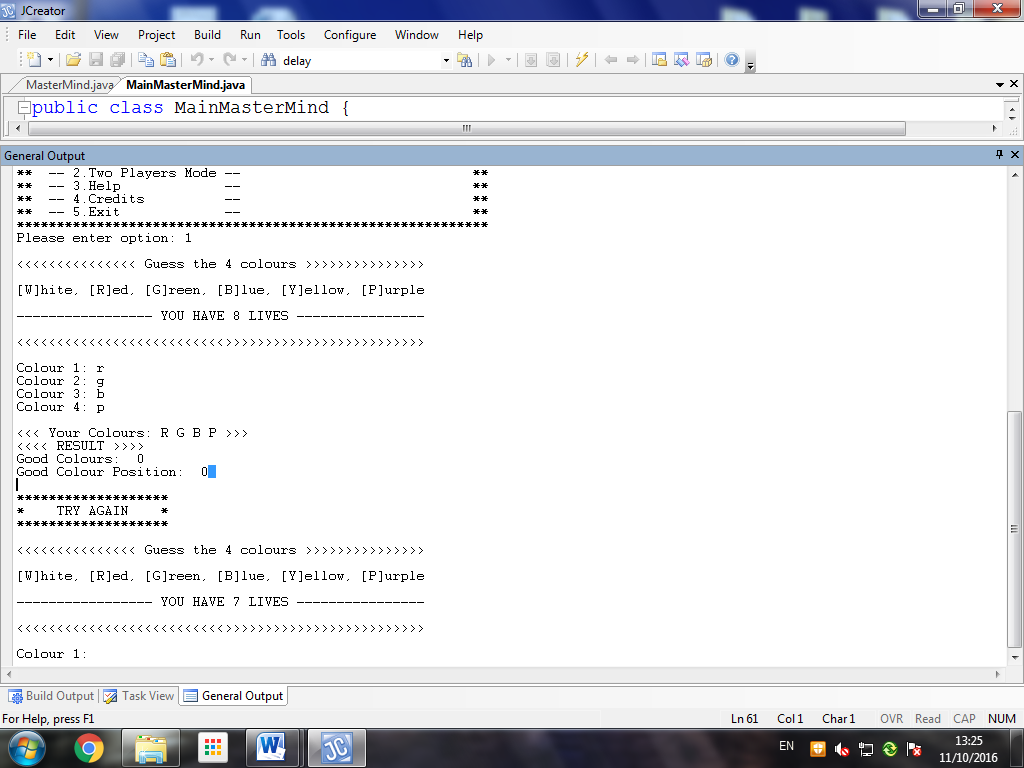
## Evidence that the solution works

* Pics showing the program running
* Caption explaining what the screenshot represents

*\* find sample screenshots below:*



Screenshot : Showing the Main Menu



Screenshot 2: During Gameplay

## Plan of Test Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Plan** | **Test Data** | **Expected Output** | **Actual Output** | **Remarks** |
| Credits option in Main Menu works | option = 4 | Credits information is shown and returns to main menu | Credits are shown and returned to main menu | Test Passed |
| Invalid Option in Main menu | option = 8 | Show Invalid Choice and return to Main Menu | Returned to Main Menu | Test Failed …to be modified |
| Invalid colour is entered during gameplay | user[inputCount]=’l’ | An Invalid message is shown and prompts the user to enter the colour again | An Invalid message is shown, and the user was prompted to enter the colour again | Test Passed |
|  |  |  |  | Test Failed …for future upgrades |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 

# Chapter 4: User Instruction

## Loading and using the program

## 

* How to install the program,
* How to run the program,
* How to recover from errors.

*\* To include screen by screen instructions using clearly labelled screenshots*

# 

# Chapter 5: Comments and Conclusions

## Limitations and Improvement

* Evaluate what you have learnt from doing this project.
* Did you enjoy doing it?
* What were your expectations and where did you arrive?
* What improvements can you include in your software?
* Mention any limitations you encountered such as time constriction, programming ability, etc.