**SPL-1 Project Report, 2019**

**Lazy Coder**

**Course: Software Project Lab I**

**Course No: SE 305**

Submitted by

**Md. Bashirul Islam**

**BSSE Roll No. : 1022**

**BSSE Session: 2017-2018**

Supervised by

**Dr Mohammed Shafiul Alam KhanDesignation: Associate Professor**

**Institute of Information Technology**

****

**Institute of Information Technology**

**University of Dhaka**

**29-05-2019**

**Table of Contents**

**1.Introduction ..................................................................................... 1**

**1.1.Background study ..................................................................... 1-2**

**1.2.Challenges ..................................................................................... 3**

**2.Project Overview .......................................................................... 3-14**

**3.User Manual .................................................................................…15**

**4.Conclusion ....................................................................................... 16**

**5.Appendix........................................................................................... 16**

**6.References ........................................................................................ 16**

**Index of Figures**

**Figure 1**: **User Interface** **…. .................................................................................... 2**

**Figure 2: C code vs natural Instruction...................................................................4**

**Figure 3:** **C code template....................................................................................…5**

**Figure 4: Error checking of variable.....................................................................6**

**Figure 5: Type mismatch checking....................................................................... 7**

**Figure 6: Semantic error..........................................................................………...8**

**Figure 7: Data Type mismatch handling..............................................................9**

**Figure 8: Operator, Operand mismatch handling.............................................10**

**Figure 9:** **operator, operand mismatch example................................................11**

**Figure 10: Bracket mismatch handling................................................................11**

**Figure 11: Bracket missing error..........................................................................12**

**Figure 12: Variable not found error handling................................................... 12**

**Figure 13: Loop statement......................................................................................13**

**Figure 14: Implementation of Loop statement....................................................13**

**Figure15: Example of Loop statement………………………………………..14**

**Figure16: Generated C code ………………………………………………….14**

**Figure17: output sample……………………………………………………….15**

**Figure18: user manual…………………………………………………………15**

**1**

1. **Introduction**

The goal of the project is to develop a software tool for the beginner programmers.This tool helps the user to learn c programming in a easier way as the user will provide instructions which is quite similar to natural language and the tool will convert it to c code. So the user will be able to see the corresponding c language instruction.

It also gives suggestion if any syntactically error occurred by the user.

For example the user may try to use a variable without declaration then the tool gives an error message to the user that “the variable is not declared yet.”I don't use any resource from online or anywhere to build this project.

**1.1 Background study**

**C Language Syntax learning:**

To implement this project I have to study more about the c language syntax & how it actually works. Then I have to design it more simple way to present it for the user. The tools present every C language instruction in a new and more easier way. For example it can be noted that for representing the “For Loop” statement I use “Make repeated statement” button and present it as “Make repeated statement for n times[” which is quite similar to natural language. So by using this tool the user can easily understand what is a loop and how a works.

**File Read and Write:**

File read and write is an important task to implement this project. For every instruction given by the user first I have to convert it proper C Language form. Then I have to find the exact place of the generating C code file to place the new generated instruction.

For reading data of files, I have to study some functions mentioned below

o Opening file

o Reading and writing

o Closing file

**2**

**Java FX:**

JavaFX is a [software platform](https://en.wikipedia.org/wiki/Computing_platform) for creating and delivering [desktop applications](https://en.wikipedia.org/wiki/Application_software), as well as [rich Internet applications (RIAs)](https://en.wikipedia.org/wiki/Rich_Internet_application) that can run across a wide variety of devices. JavaFX is intended to replace [Swing](https://en.wikipedia.org/wiki/Swing_(Java)) as the standard [GUI](https://en.wikipedia.org/wiki/Graphical_User_Interface) library for [Java SE](https://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition)

One of the important perspective of this project is to give a better & more easier representation of the normal C language instruction. So to represent in a better I have to design a graphically user interface using Java FX.



Figure1: User Interface

**Errors in C Language:**

Error is an illegal operation performed by the user which results in abnormal working of the program.

Programming errors often remain undetected until the program is compiled or executed. Some of the errors inhibit the program from getting compiled or executed. Thus errors should be removed before compiling and executing

There are many types of errors in C language. Among them this tool will provide error notification if any syntax error occurs. So I have to study about possible errors for this instruction.

3

**1.2 Challenges:**

As it is my first software project, I have to face a lot of challenges. I have to handle a large code for the first time.

Few more challenges I have faced-

* Handling syntax error of the instruction is a huge challenge for me, as I tried to handle these errors case by case. I have to find out all possible errors for a certain instruction.
* Finding out a better way to represent each C language instruction.
* Designing the graphically user interface such that the user can easily understand all functions of the tools.
* Read the C file for every instruction and write that instruction in the correct position.
* Handling large code for the first time.
* Learning and implementing JavaFX for the first time.

1. **Project Overview :**

I have divided my whole project into three different phases. They are

* Understanding the project & challenges.
* Implementing some C instruction like variable declaration, assign variable, doing some operation, making loop etc.
* Create a graphically user interface for the tool.
  1. **Understanding the project & challenges:**

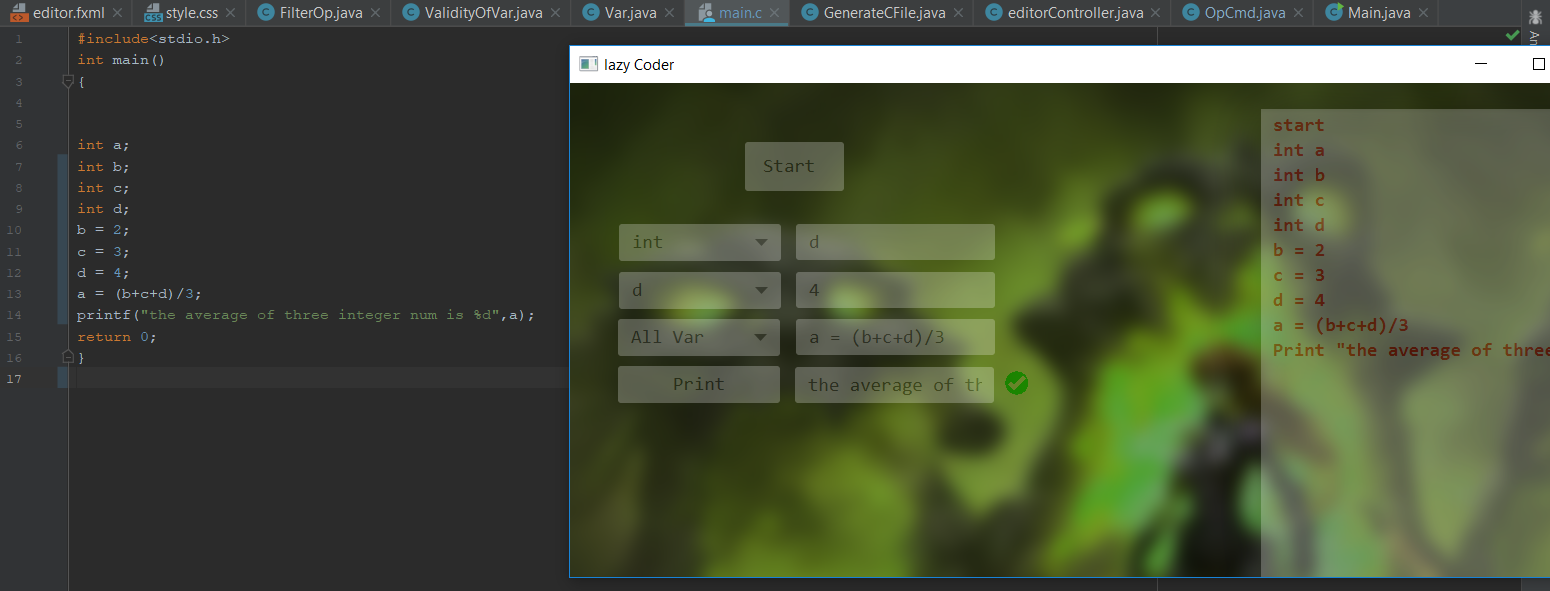
Understanding the project was the most challenging part for me. I have to study about the all possible errors and its solution. Most of the time I had to spend behind solving this problem. I had to filter every instruction given by the user through checking the all possible errors. This phases took almost 40% of my total working time for this project.

4

* 1. **Implementation:**
* **C Code Generator**Generate c code from some specific instruction which is quite similar to natural language.

The process I have followed**:**

* taking instructions from user line by line
* manipulate it through some specific process
* convert it to corresponding c language
* read the c file and write the corresponding c code instruction on correct position



**Figure2: C code vs natural Instruction**

**5**

* As C is a structured programming language, initially creates a template like this:

#include<stdio.h>

int main()

{

return 0;

}

public void startIt() throws IOException  
{  
 fout = new FileOutputStream(new File("main.c"));  
 fout.write("".getBytes());  
 String start = "#include<stdio.h>\n"  
 + "int main()\n"  
 + "{\n\n\n"  
 + "return 0;\n"  
 + "}\n";  
  
 fout.write(start.getBytes());  
}

Figure3: C code template

In the next step, for every instruction after converting it to a C instruction, it placed exactly before the “return 0;”. And that’s why I had to read the whole .c file for each and every instruction and after placing it to the right position I had to rewrite it to the file.

* For declaring a variable name, I have to consider some rules like : it can’t be a key word in the C language and had to ensure that it had not been used before. If the user tries to do something like that I had to manage an error notification in the tool.

To solve this problem I used an array list for the variables are already used and store all the keyword of C language in a text file name Key.txt. For every variable declaration I checked if the text file and array list contains the name.

6

public boolean validity(String variableName) throws IOException {  
 int tmp;  
 String addit = "";  
 fin = new FileInputStream(new File("Key.txt"));  
 while(fin.available()!=0)  
 {  
 tmp = fin.read();  
 addit = addit+(char)tmp;  
 }  
 char[] stringToCharArray = addit.toCharArray();  
 String key = "";  
 for (int i =0; i<=addit.length();i++){  
 if(i == addit.length()){  
 if(key!=null) {  
 Keyes.add(key);  
 key = "";  
 break;  
 }  
 break;  
 }  
 if(stringToCharArray[i]==' '){  
 if(key!=null) {  
 Keyes.add(key);  
 key = "";  
 }  
 }  
 else key = key + stringToCharArray[i];  
 }  
 if(variableName.contains(" ")){  
 editorController.*error* = "variable Name cannot contain space!!! Invalid Variable!!!!";  
 return false;  
 }  
 if(Keyes.contains(variableName)){  
 editorController.*error* = "Invalid Variable Name!!! This is a keyWord/operator in c/c++ language!!!";  
 return false;  
 }  
 else {  
 Var var1 = new Var();  
 for(int i =0;i<arrayListOfVar.size();i++){  
 var1 = arrayListOfVar.get(i);  
 if(variableName.equals(var1.VarName)){  
 editorController.*error* = "variable name already exist!!!!";  
 return false;  
 }  
 }  
 return true;  
 }

Figure4: Error checking of variable

7

* To assign value in a variable first I had to ensure it that the variable already declared before. Then I have to check the inserted value either matches with the variable type or not.

Var var1 = new Var();  
boolean mark = false;  
for (int i = 0; i < arrayListOfVar.size(); i++) {  
 var1 = arrayListOfVar.get(i);  
 if (varName.equals(var1.VarName)) {  
 String value = valueHolder.getText();  
 if(var1.VarType.equals("double")){  
 for(int x =0;x<value.length();x++){  
 if(value.charAt(x) == '.'){  
 try {  
 if(!(value.charAt(x+1)>='0' && value.charAt(x+1)<='9'))  
 *error* = "Type mismatch!!!";  
 ValueError.setText(*error*);  
 *error* = "";  
 return;  
 }catch (Exception e){  
 *error* = "Invalid value!!!";  
 ValueError.setText(*error*);  
 *error* = "";  
 return;  
 }  
 }  
  
 else if(!((value.charAt(x)>='0' && value.charAt(x)<='9') || (value.charAt(x) == '.' ))){  
 *error* = "Type mismatch!!!";  
 ValueError.setText(*error*);  
 *error* = "";  
 return;  
 }  
 else {  
 try {  
 c.assignVar(varName, value);  
 *mainOutput* = *mainOutput* + varName + " = " + value + "\n";  
 output.setText(*mainOutput*);  
 var1.setValue(value);  
 ValueError.setVisible(false);  
 } catch (Exception e) {  
  
 }

}

}

Figure5: Type mismatch checking

8

**Error Handling:**

* This tool is developed for only for the beginner developers. So most the facilities/operations of C language is not implemented. Rather I tried to make it easier the basic operation to the user.

To implement this I had to consider a lot of errors. Among them :

* **Semantic errors :** This error occurs when the statements written in the program are not meaningful to the compiler.

// C program to illustrate

// semantic error

void main()

{

   int a, b, c;

   a + b = c; //semantic error

}

error: lvalue required as left operand of assignment

a + b = c; //semantic error

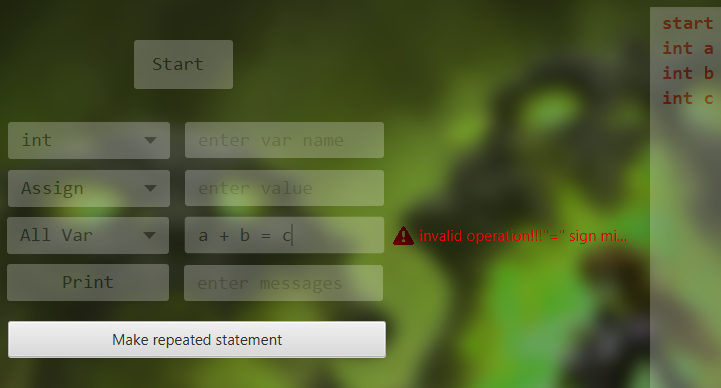


Figure6: Semantic error

**9**

* **Data type mismatch :** This may occur when L value and R value type doesn’t match**.**

// C program to illustrate

// Data type mismatch

void main()

{

   int a, b;

char c;

   c = a + b; // Data type mismatch

}

error: L value and R value type doesn’t match**.**

c = a + b; // Data type mismatch

or,

// C program to illustrate

void main()

{

   int a;

   c = ‘a’; // Data type mismatch

}

error: L value and R value type doesn’t match**.**

c = ‘a’; // Data type mismatch

public boolean checkType(){  
 if(arrayListOfVariableInOP.size()==1){  
 return true;  
 }  
 String type = null;  
  
 type = arrayListOfVariableInOP.get(0).VarType;  
 for(int i = 1; i<arrayListOfVariableInOP.size();i++){  
 if(type.equals(arrayListOfVariableInOP.get(i).VarType)){  
 x = true;  
 }  
 else x= false;  
 }  
 if(x == true){  
 return x;  
 }  
 else {  
 System.*out*.println("variable type missMatch!!!");  
 editorController.*error* = "variable type missMatch!!!";  
 return x;  
 }  
}

Figure7: Data Type mismatch handling

10

* **Operator, operand mismatch :**

// C program to illustrate

void main()

{

   int a, b, c;

   c = (a b)/5; // Operator, operand mismatch

}

error: operator missing between a & b**.**

c = (a b)/5; // Operator, operand mismatch

To solve this problem I had used different array list of operator, operand. Then ensured that the sequence is exactly operand, operator, operand in this format.

public void validityCheck() throws IndexOutOfBoundsException {  
  
 String x1 = arrayListOfOp.get(0);  
 String x2 = arraylistOfVarName.get(0);  
 if(x1!=x2){   
 editorController.*error* = "something went wrong!!!variable mismatch!!!";  
 return;  
 }  
 int j =0;  
 String s1 = "=";  
 String s2 = null;  
 try{  
 s2 = arrayListOfOp.get(1);  
 }  
 catch (Exception e){  
 editorController.*error* = "Invalid Operation!!!";  
 return;  
 }  
 if(s1.equals(s2)){  
  
 for(int i = 0; i<arrayListOfOperator.size();i++){  
 j = 2\*i+1;  
 if(arrayListOfOperator.get(i)!=arrayListOfOp.get(j)){  
 //System.out.println("invalid operation!!!operator missing....");  
 editorController.*error* = "invalid operation!!!operator missing....";  
 return;  
 }  
  
 }  
 String a1 = arrayListOfOperator.get(0);  
 String a2 = arrayListOfOperator.get(arrayListOfOperator.size()-1);  
 if((arrayListOfOp.get(0)!= a1) && (arrayListOfOp.get(arrayListOfOp.size()-1)!=a2)){  
 //System.out.println("----valid operation---");  
 x = true;  
 }  
 else{  
 editorController.*error* = "invalid operation!!!";  
 return;  
 }  
  
 }  
  
 else{  
 editorController.*error* = "invalid operation!!!\"=\" sign missing!!!";  
 return;  
 }  
}

Figure8: Operator, Operand mismatch handling

11

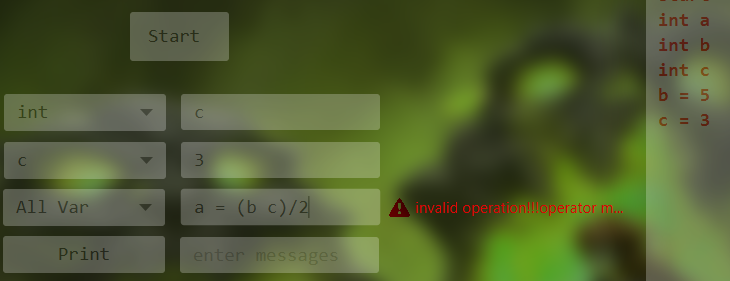


Figure9: operator, operand mismatch

* **Bracket mismatch :** opening or closing bracket missing.

For solving this problem I use stack for counting bracket.

public boolean brucketMatching(String OpCmd)throws EmptyStackException {  
 Stack<String> st = new Stack<String>();  
 String br = "";  
 String operation = "";  
 char[] stringToCharArray = OpCmd.toCharArray();  
 for(int i=0;i<OpCmd.length();i++){  
 if(stringToCharArray[i] == '('){  
 br = br + stringToCharArray[i];  
 st.push(br);  
 br = "";  
 }  
 if(stringToCharArray[i] == ')'){  
 if(st.empty()) {  
 editorController.*error* = "invalid operation!!! \" opening or closing Bracket missing!! \" ";  
 return x;  
 }  
 else  
 st.pop();  
 }  
 else if((stringToCharArray[i]!= '(') && (stringToCharArray[i]!=')')) operation = operation + stringToCharArray[i];  
 if(st.empty()){  
 msg = operation;  
 filter();  
 }  
 else editorController.*error* = "invalid operation!!!";  
 return x;  
}

Figure10: Bracket mismatch handling

12

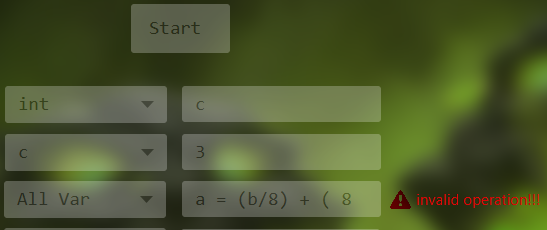
****

Figure11: Bracket missing error

* **Print a variable without declaration**: In the print instruction user may try to print a variable without declaring it.

So, first I have to ensure that the variable is assigned or not.

public void filter(String msg) throws Exception {  
 char[] stringToCharArray = msg.toCharArray();  
 for(int i =0;i<msg.length();i++){  
 if(stringToCharArray[i]=='('){  
 String var="";  
 while(true){  
 i++;  
 if(stringToCharArray[i]==')'){  
 arraylistOfVarName.add(var);  
 var = "";  
 break;  
 }  
 if(!(stringToCharArray[i]=='(' || stringToCharArray[i]==')'))  
 var = var + stringToCharArray[i];  
 }  
 }  
 }  
 for(int i =0;i<arraylistOfVarName.size();i++)  
 {  
 String s = (String) arraylistOfVarName.get(i);

String VarName = " "+s;  
 int pos = getPosition(VarName);  
 if(pos<0){  
 editorController.*error* = "Variable not Found!!! please dclare the variable first---";  
 return;  
 }  
 getVarType(pos);  
 }  
 CreateString(msg);  
}

Figure12: Variable not found error handling

13

* **Implementing the Loop statement:** The challengingpart of this portion is to remind it when the loop terminate and more complexity was arise when I want to implement nested loop.

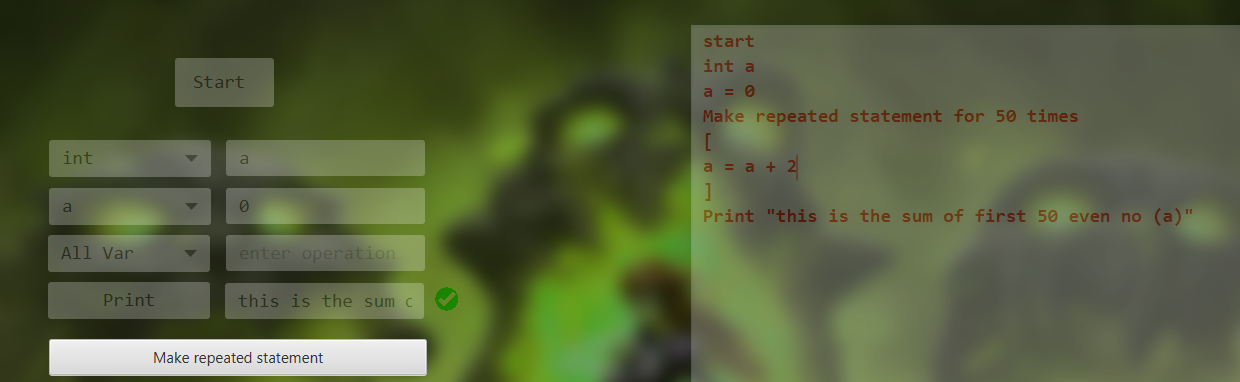


Figure13: Loop statement

public void makeLoop(String command,String iteration) throws Exception {  
 String p = "";  
 if(command.equals("makeLoop")){  
 String i = iteration;  
 editorController.*s*.push("{");  
 String marker = String.*valueOf*(editorController.*s*.size());  
 p = "for(int iteration\_" + marker + "=0 ; iteration\_" + marker + " <" + i + "; iteration\_" + marker + "++ )\n{";  
 FileWrite(p);  
 }  
 else if(command.equals("exit")){  
 p = "\n}";  
 editorController.*s*.pop();  
 FileWrite(p);  
 }  
}

Figure14: Implementation of Loop statement

Here is example of iterative statement in the next page. Here I tried to draw a squire shape using the tool.

14

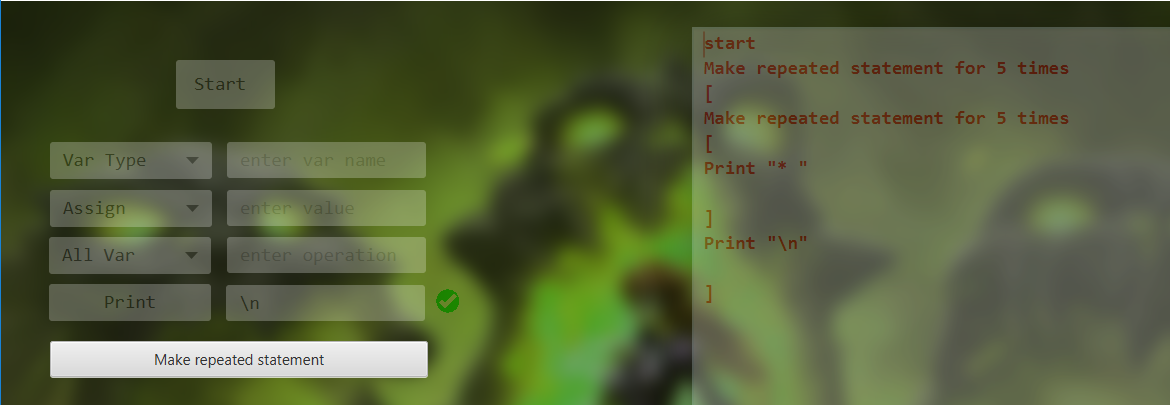
****

Figure15: Example of Loop statement

**Generated C code:**

#include<stdio.h>  
int main()  
{  
for(int iteration\_1=0 ; iteration\_1 <5; iteration\_1++ )  
{  
for(int iteration\_2=0 ; iteration\_2 <5; iteration\_2++ )  
{  
printf("\* ");  
  
}  
printf("\n");  
  
}  
return 0;  
}

Figure16: Generated C code

**Output:**

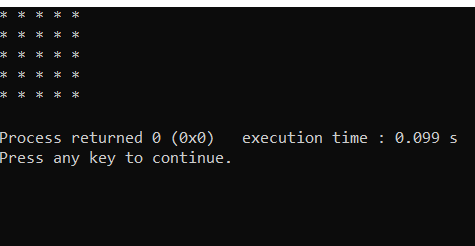
****

Figure17: output sample

**15**

1. **User manual :**

For starting writing code the user must have to press the start button. By clicking it a C code template will create in the backend.

* Variable declaration: User can choose a type from the choice list and can give a suitable name in the text field.
* Variable assign: User can choose a variable from the choice list and can give a suitable value in the text field in the right side of the choice list.
* Create operation: user can insert any suitable operation here.
* Print: For printing something user has to click the print button at first. Then the text field in the right side will be enable. For printing a variable user has to enclose it by the “()” bracket.
* For making loop user has to click in the “make repeated statement”. Then a pop up box will arise, where the user can insert the iteration no.
* After giving an iteration no a “exit loop” button will arrive by clicking it user can exit from a loop.

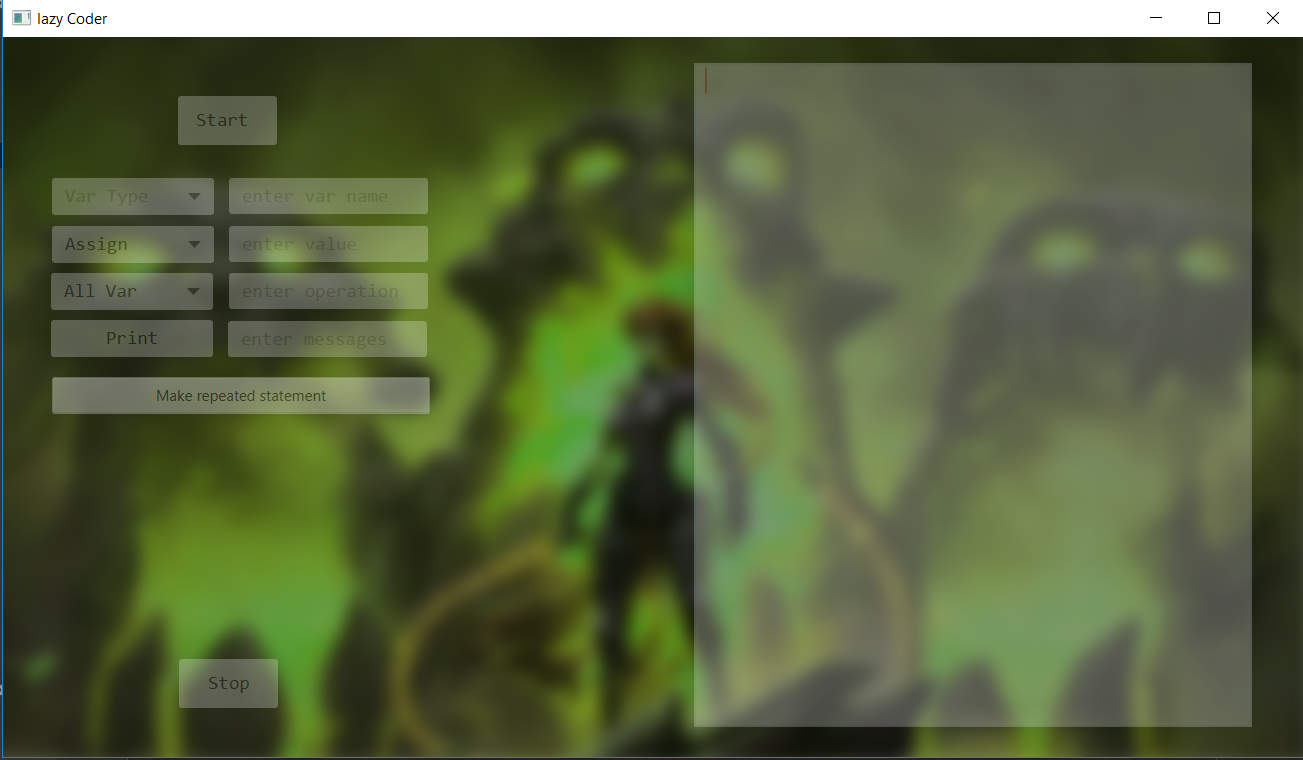
****

Figure18: user manual

**16**

# 4. Conclusion

Implementation of this project helps me to understand a new topic how to read and write file in java. I have to learn C language syntax. The project also helps me to improve my coding skill and I have learned to handle large code for the first time. I have learnt to use java in different scenario and have learnt to use java FX to create graphical user interface. I hope it will help me to deal with difficulties in future. This project was quiet challenging and I gained a lot of experience from it. I want to thank my supervisor and other respectable teachers for guiding me a lot during this project.

# 5. Appendix

I want to make it a fully intelligent tool. To complete it a lot of thins yet to be implanted.

Among them….

* Conditional statement(if, if else)
* User defined function
* Array
* Login section
* Code viewer

So in future, so in future I want to implement all of these feature to complete the project.

**6. Reference**

1. <https://www.tutorialspoint.com/cprogramming/c_file_io.html>
2. <https://www.tutorialspoint.com/java/java_files_io.htm>
3. <https://code.makery.ch/library/javafx-tutorial/part1/>