**USER MANUAL**

Welcome to CSE HELPER. An assistive application for flowchart to code converter, flow chart tracing and interactive visualization of various data structures for students to make the learning process easier.

This portable application needs JDK 7 installed and the running java platform should be JDK 1.7. Upon double clicking the executable jar file (CSE310\_Project.jar), the application will start.

The two parts of this application, 1) flowchart part is accessed by clicking the CSE110 button and 2) data structure visualization part is accessed by CSE220 button.

**CSE110 or Flowchart section**:

CSE110 section consists of 3 parts-

1. On the upper side, the buttons for different flowchart symbols to make a complete flowchart.

2. On the left side, the panel that flowchart is created and shown.

3. On the right side, the code for corresponding symbols is generated and instantly shown to see the flow of the flowchart.

**1. The Buttons for flowchart symbols**:

There are nine buttons for flowchart symbols such as Start, Process by input, Process, Conditional, Loop, Output, End, Compile and run, and New.

Upon clicking each button, the left panel shows the corresponding symbols by maintaining the flow with previous symbols and the right panel shows the code generated by clicking the buttons.

**Start Button**: Start button starts the flowchart and is disabled until the New Button is clicked for a new flowchart.

**Process by Input Button:** This button is used to declare new variable with an initial value assigned to it. Upon click, a small window will pop up asking for the type of the variable (Strings are not allowed) and then another pop up window will take the name of the variable and finally the initial value will be taken from the user.

If the variable type and value do not match, an error will occur during compile time. **User must enter only what is asked otherwise the generated code will have incorrect syntax and will cause a compile time error. No ‘;’ or opening or closing brackets are needed.** Examples of inputs are provided on the pop up window.

**Process Button:** Process button is used for changing the value of already declared value. If the variable entered is not declared before with the Process by Input, a compile time error will occur.

Examples: a == true, a == 10

**Conditional Button:** This button is used for creating an IF/ELSE block. One conditional button creates only one IF/ELSE block with **MUST** two symbols (output or process) in both the IF and ELSE part. No Loop or new variable can be declared inside IF/ELSE block.Putting a conditional inside another will cause unwanted behaviours.

**Loop Button:** This button creates a while loop with two symbols (one must be a process to end the loop at a certain condition). **WITHOUT PROPER INCREMENT OR DECREMENT AN INFINITE LOOP MIGHT OCCUR.** No Conditional or new variable can be declared in the loop.

**Output Button:** This button prints what is entered during input taking. It puts the input taken from the pop up window into System.out.println (//input goes here). An error might occur if inputs are not valid.

**End Button:** This button ends the flowchart and also the generated code.

**Compile & Run Button:** This button compiles and runs the generated code and depending on successful compilation prints the output or shows error message with some possible error causing elements.

**New Button:** New button cleans the left panel and right panel for new flowchart.

Buttons are enabled and disabled sometimes to control the flowchart.

**CSE220 or Data Structure Visualization Section**:

On the top, there are four tabs – Array, LinkedList, Stack and Queue. Under the Array tab, there are two tabs for Linear and Circular Array.

Like flowchart, buttons are enabled and disabled in all data structures to avoid unwanted behaviours. All the buttons have tool tips to know each one’s function. In all data structures except LinkedList, only ten elements can be manipulated. For LinkedList five nodes excluding head node can be manipulated. All data structures values are Integers (int), entering other variable type will not work.

**Array:**

Linear Array: This tab is for linear array visualization.

Circular Array: This tab is for circular array visualization.

**LinkedList:** This tab is for LinkedList visualization. A head-referenced singly LinkedList is used.

**Stack:** This tab is for stack visualization. Provided functions are push, Pop, peek, and search.

**Queue:** This tab is for queue visualization. Provided functions are enqueue, dequeue, peek, and search.

All data structure tabs have pause and resume button to pause the task and resume it to continue.