# SCREEN CAPTURE ONE

**SHOW VARIABLES WINDOW  
A screenshot of a computer

Description automatically generated**

# TABLE ONE

|  |
| --- |
| Question: What is the Eclipse keyboard shortcut for toggling a breakpoint? |
| Answer: Ctrl + Shift + B |
| Question: What is the difference between “Step-Over”, and “Step-Into”, and “Step-Return”? |
| Answer:  Step-Over – Executes each line of code without going into the call. Step-Into – Executes each line of code and goes deeper into each call. Step-Return – Returns to the line that called for the current line of code. |
| Task: Practice tracing through the DebugStar sample program. It is ok if you don’t understand all of the java code; but you should be able to trace the order in which statements are executed.  Based on your best understanding of the program, provide a list of methods that are called when the program executes (from start to end, in order of being called). You can skip library methods (like println, for example).  HINT: Use a combination of “Step-Into” “Step-Over” and “Step-Return”. Use the “Stack Trace” window. |
| List of Methods (in order of call) below. Please use the fully qualified name, eg. “DebugStar.run(String, int, int). Use the stack view to help you.  DebugStar.main(String[]) //line 6 – first line that is executed in the main method  DebugStar.run(String, int, int)  DebugStar.getOperation(String)  Add.perform(int, int)  DebugStar.main(String[]) //line 7 – second line that is executed in the main method  DebugStar.run(String, int, int)  DebugStar.getOperation(String)  Subtract.perform(int, int)  DebugStar.main(String[]) //line 8 – third line that is executed in the main method  DebugStar.run(String, int, int)  Factorial.perform(int) |

2. The Debug Challenge

1. Provide screen capture of original code (with line numbers)

SCREEN CAPTURE: Original Code with line numbers

|  |
| --- |
|  |

2. Provide an error log table (such as the one below) indicating error details (line number, type of error, and explain error and show correction).

|  |  |  |  |
| --- | --- | --- | --- |
| **Line**  **Number** | **Type of error (compile-time, run-time, or**  **logical)** | **Description** | **Correction** |
| 26 | compile error | Missing “;” in the statement | int n; |
| 27, 31 | compile error | variable n is not initialized. | int n = 0; |
| 27 | logic error | when n is assigned zero, it does not satisfy the condition to call the while loop. | while (n < 2) |
| 43 | logic error | Wrong index: Adds the previous object from the array list to itself, instead of adding the previous 2 objects. | f.add(f.get(i - 2) + f.get(i - 1)); |
| 51 | run-time error | loop condition allows an out of bounds index to be applied to the array causing a run-time error. | while (i < fiboList.size()) |
| 50 | logic error | this prints the array elements from index 2 missing out the first 2 elements (index 0 and 1). | int i = 0; |
| 52 | logic error | this prints all the array elements without any formatting and does not look like the required output. | System.out.print(fiboList.get(i) + ", "); |

3. Provide screen capture of fixed code (with line numbers), and sample run, using n=10:

SCREEN CAPTURE: Fixed code with line numbers

|  |
| --- |
|  |