1. Getting Familiar with Debugging in Eclipse

Screen Capture One:

Table

Description automatically generated

Table One:

|  |
| --- |
| Question: What is the Eclipse keyboard shortcut for toggling a breakpoint? |
| Ctrl + Shift + B |
| Question: What is the difference between “Step-Over”, and “Step-Into”, and “Step-Return”? |
| Step-Over: is used to execute individual lines of code.  Step-Into: is used to execute individual lines in a method.  Step-Return: tell the debugger to not execute the method. |
| 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[])  DebugStar.main(String[])  DebugStar.main(String[])  Factorial.perform(int)  DebugStar.run(int)  DebugStar.main(String[]) |

1. The Debug Challenge

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

SCREEN CAPTURE: Original Code with line numbers

Screen Capture of Original Code:

A picture containing text

Description automatically generated

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 |
| 27 | Compile-time error | The local variable n may not have been initialized | Initialize variable n:  System.out.println("Please enter a number: ");  int n = scanner.nextInt(); |
| 31 | Compile-time error | The local variable n may not have been initialized | Initialize variable n above |
| 27 | Logical error | Infinity loop of “Enter a number bigger than 2” because “n > 2” is the correct condition, this is the condition for the incorrect condition to re-enter the number. | Fix the condition:  while (n < 2) |
| 40 | Logical error | Wrong formula for the Fibonacci Sequence. | Fix the formula:  f.add(f.get(i - 1) + f.get(i - 2)); |
| 47 | Logical error | The Fibonacci Sequence starts from number 0, so the loop must start from 0. | int i = 0; |
| 48 | Run-time error | i has to be smaller than the size of the fiboList or the index will be out of bound of the array (the last index we can access is size of the array minus 1) | i < fiboList.size() |
| 32 |  | Scanner is never closed | Add: scanner.close(); |

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

SCREEEN CAPTURE: Fixed code with line numbers

Graphical user interface, text, application

Description automatically generated