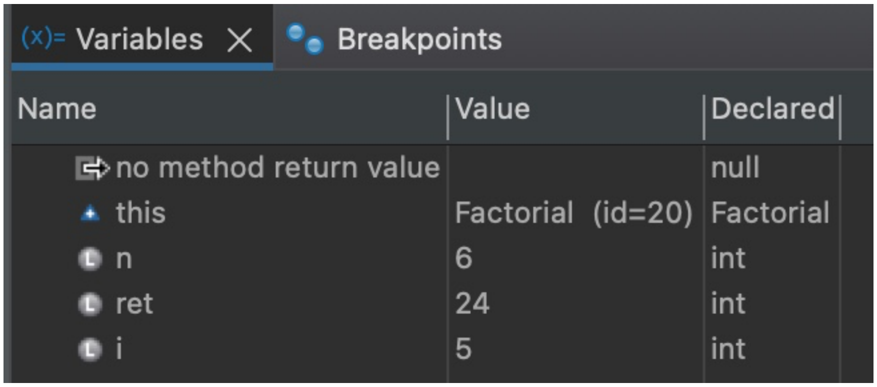
1. Getting Familiar with Debugging in Eclipse

SCREEN CAPTURE ONE

VARIABLES WINDOW:



Name	Value	Declared
no method return value		null
this	Factorial (id=20)	Factorial
n	6	int
ret	24	int
i	5	int

TABLE ONE

Question: What is the Eclipse keyboard shortcut for toggling a breakpoint

Answer:

CTRL + Shift + B (Windows)
CMD + Shift + B (MacOS)

Question What is the difference between "Step-Over", "Step-Into", and "Step-Return"?

Answer:

- Step-Over – Execute the function and you can see inside the function and see how the function is executing line by line. Then returns you to the next line right after the function call.
- Step-Into – Execute the function call and returns the result but you cannot see how the function was executed.
- Step-Return – Execute until the end of the current function, stop after calling line.

Task: Practice tracing through the DebugStar sample program.

It is ok if you don't understand all 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.

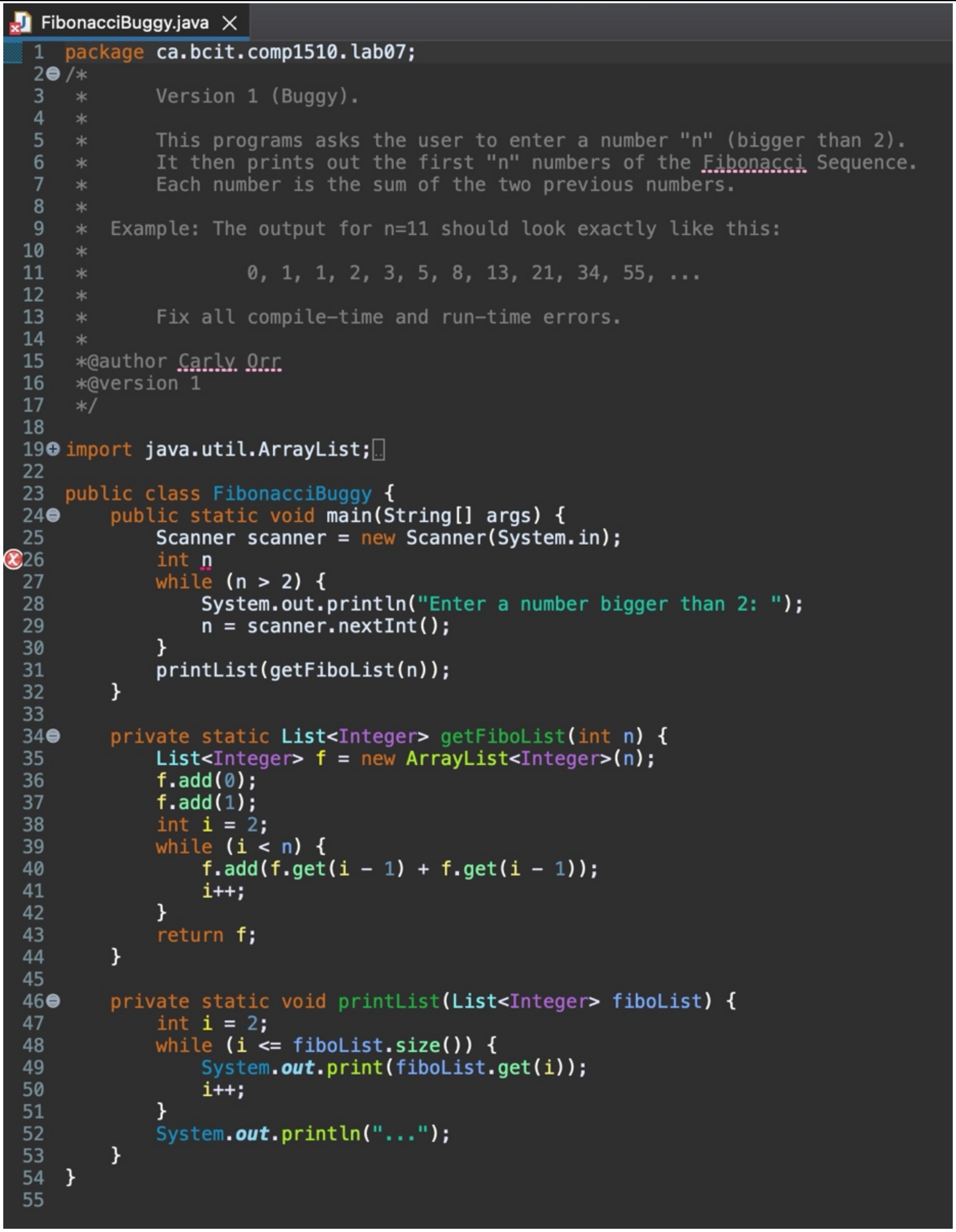
List of Methods (in order below). Please use the fully qualified name, eg. “Debugstar.run(String, int, int)”. Use the stack view to help you.
Answer:

1. DebugStar.main(String[]) line:8
2. DebugStar.run(int) line: 33
3. Factorial.perform(int) line: 62
4. Factorial.perform(int) line: 61
5. Factorial.perform(int) line: 64
6. DebugStar.run(int) line: 33
7. DebugStar.run(int) line 34
8. DebugStar.run(int) line: 35
9. DebugStar.main(String[]) line: 9

2. The Debug Challenge

You are given a source code for a program that is “BUGGY” called FibonacciBuggy. Your job is to find out why, using the skills that you have acquired so far.

1. SCREEN CAPTURE: Original Code with line number

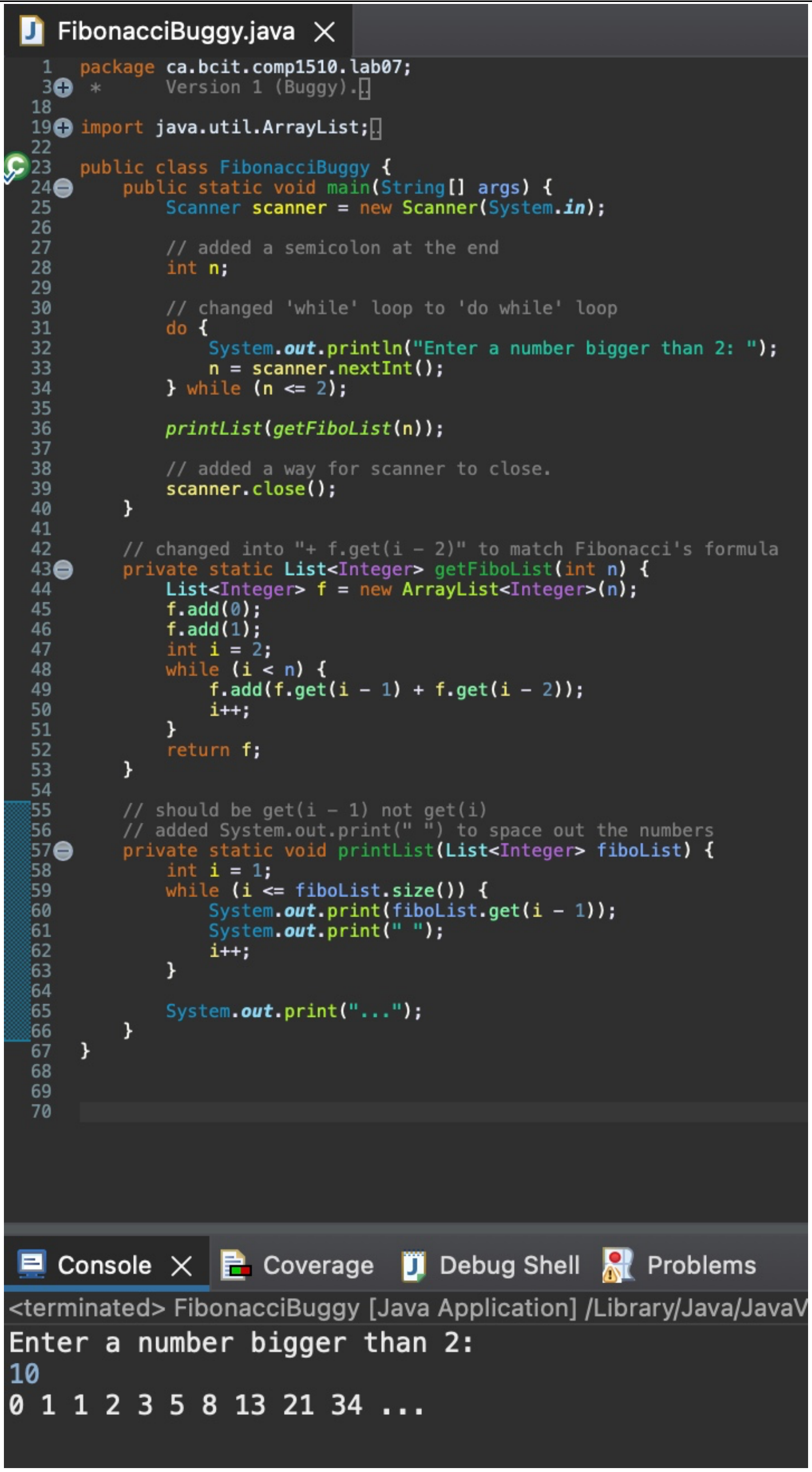


```java
package ca.bcit.comp1510.lab07;
/*
 *      Version 1 (Buggy).
 *
 *      This programs asks the user to enter a number "n" (bigger than 2).
 *      It then prints out the first "n" numbers of the Fibonacci Sequence.
 *      Each number is the sum of the two previous numbers.
 *
 *  Example: The output for n=11 should look exactly like this:
 *
 *              0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
 *
 *      Fix all compile-time and run-time errors.
 *
 *@author Carly Orr
 *@version 1
 */

import java.util.ArrayList;

public class FibonacciBuggy {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n
        while (n > 2) {
            System.out.println("Enter a number bigger than 2: ");
            n = scanner.nextInt();
        }
        printList(getFiboList(n));
    }

    private static List<Integer> getFiboList(int n) {
        List<Integer> f = new ArrayList<Integer>(n);
        f.add(0);
        f.add(1);
        int i = 2;
        while (i < n) {
            f.add(f.get(i - 1) + f.get(i - 1));
            i++;
        }
        return f;
    }

    private static void printList(List<Integer> fiboList) {
        int i = 2;
        while (i <= fiboList.size()) {
            System.out.print(fiboList.get(i));
            i++;
        }
        System.out.println("...");
    }
}
```

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

Line(s)	Type of error (compile-time, run-time, or logical)	Description	Correction
28	Compile-time error	The local variable declaration statement needs to end with a semicolon.	int n;
31 - 34	Compile-time	Changed the while loop into a do-while loop	do { } while
39	Compile-time error	Closed the scanner	scanner.close()
49	Run-time error	Changed the second get method to (i – 2), so the while loop matches the Fibonacci formula	(i – 2)
60	Logical error	Changed get(i) into get(i – 1)	Get(i – 1)
61		Added empty space in between numbers using System.out.print(“ “);	System.out.print(“ “);

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



```java
package ca.bcit.comp1510.lab07;
 *      Version 1 (Buggy).

import java.util.ArrayList;

public class FibonacciBuggy {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // added a semicolon at the end
        int n;

        // changed 'while' loop to 'do while' loop
        do {
            System.out.println("Enter a number bigger than 2: ");
            n = scanner.nextInt();
        } while (n <= 2);

        printList(getFiboList(n));

        // added a way for scanner to close.
        scanner.close();
    }

    // changed into "+ f.get(i - 2)" to match Fibonacci's formula
    private static List<Integer> getFiboList(int n) {
        List<Integer> f = new ArrayList<Integer>(n);
        f.add(0);
        f.add(1);
        int i = 2;
        while (i < n) {
            f.add(f.get(i - 1) + f.get(i - 2));
            i++;
        }
        return f;
    }

    // should be get(i - 1) not get(i)
    // added System.out.print(" ") to space out the numbers
    private static void printList(List<Integer> fiboList) {
        int i = 1;
        while (i <= fiboList.size()) {
            System.out.print(fiboList.get(i - 1));
            System.out.print(" ");
            i++;
        }

        System.out.print("...");
    }
}
```

```
<terminated> FibonacciBuggy [Java Application] /Library/Java/JavaV
Enter a number bigger than 2:
10
0 1 1 2 3 5 8 13 21 34 ...
```