

Mike Harvey

CSCI165 – Debugging lab

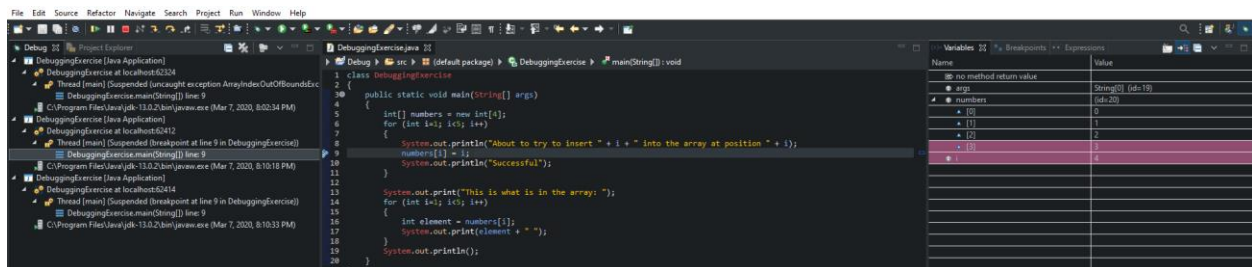
3/7/2020

DebuggingExercise.java:

Output and Error messages on compile and run:

```
<terminated> DebuggingExercise [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Mar 7, 2020, 7:51:29 PM)
Exception in thread "main" About to try to insert 1 into the array at position 1
Successful
About to try to insert 2 into the array at position 2
Successful
About to try to insert 3 into the array at position 3
Successful
About to try to insert 4 into the array at position 4
java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
    at DebuggingExercise.main(DebuggingExercise.java:9)
```

Ran program in debugger with breakpoint at line 9 and instructions to watch the numbers array. The resulting output shows the discrepancy of the array starting at 0 and running through 3 for the indexes of the 4 members and that i can get assigned values higher than that range.



The screenshot shows an IDE with the following components:

- Editor:** Displays the source code of `DebuggingExercise.java`. The code is as follows:


```

1 class DebuggingExercise
2 {
3     public static void main(String[] args)
4     {
5         int[] numbers = new int[4];
6         for (int i=0; i<4; i++)
7         {
8             System.out.println("About to try to insert " + i + " into the array at position " + i);
9             numbers[i] = i;
10            System.out.println("Successful");
11        }
12
13        System.out.print("This is what is in the array: ");
14        for (int i=0; i<4; i++)
15        {
16            int element = numbers[i];
17            System.out.print(element + " ");
18        }
19        System.out.println();
20    }
21 }
22
23
24
```
- Variables Window:** Shows the current state of variables.

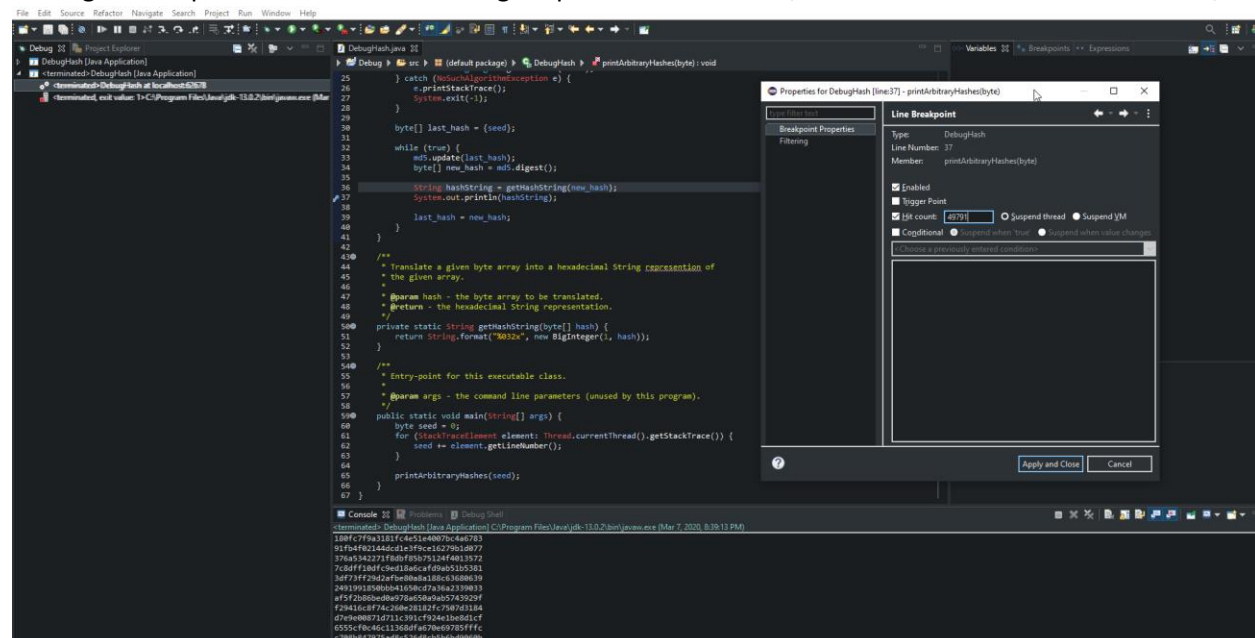
Name	Value
args	String[0] (id=19)
numbers	(id=20)
[0]	0
[1]	1
[2]	2
[3]	3
i	3
- Console Window:** Shows the output of the program.


```

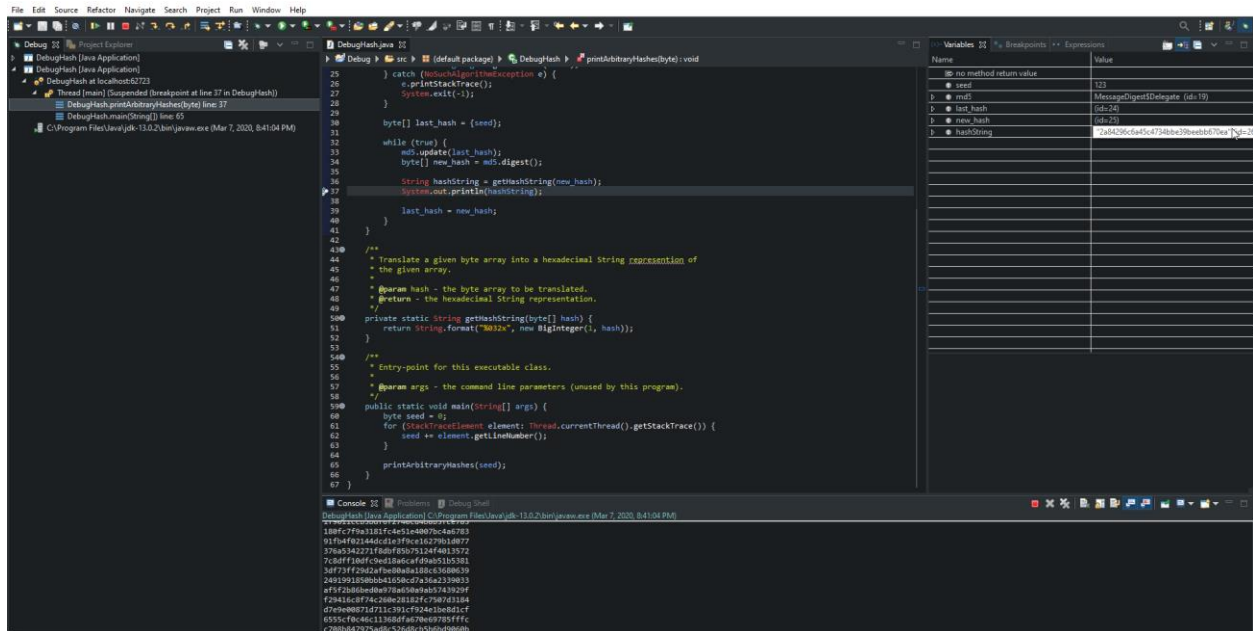
About to try to insert 0 into the array at position 0
Successful
About to try to insert 1 into the array at position 1
Successful
About to try to insert 2 into the array at position 2
Successful
About to try to insert 3 into the array at position 3
Successful
This is what is in the array: 0 1 2

```

Setting a breakpoint at line 37 and setting it up to break at the 49,791st time that line is encountered.;



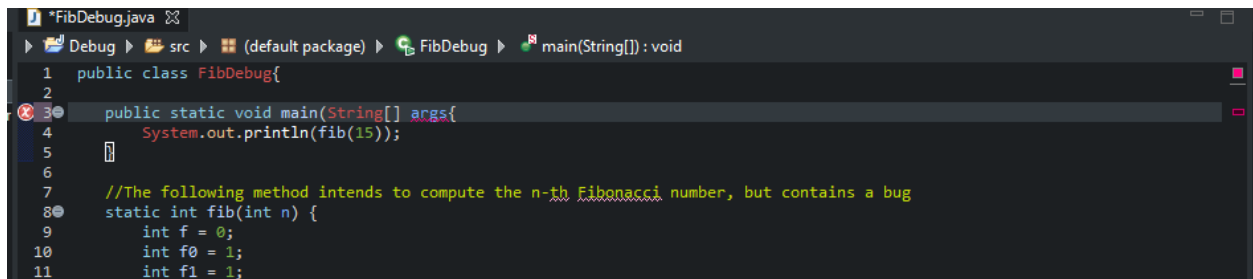
Results of running with the breakpoint containing a hit count:



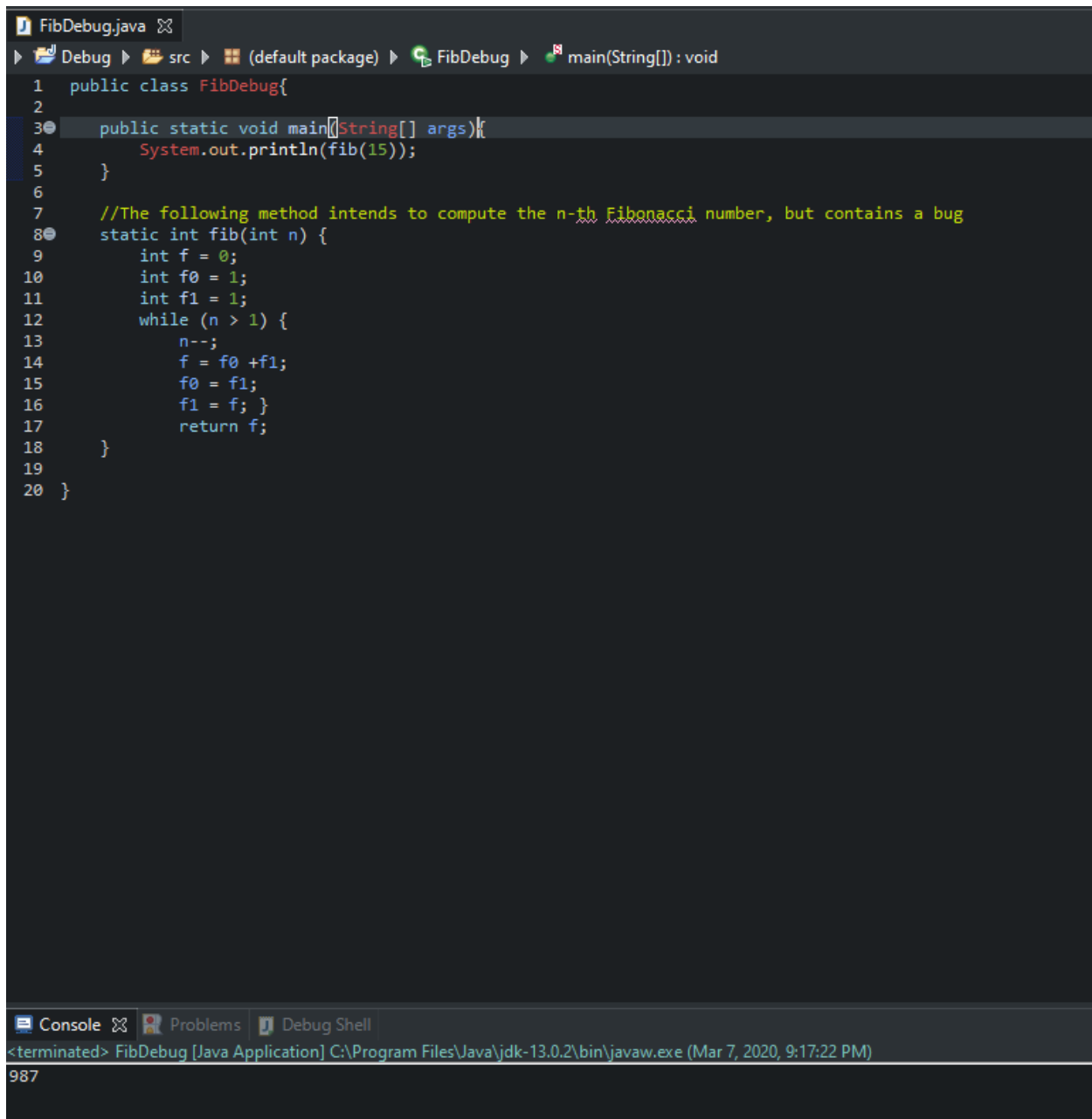
The hash value retrieved was 2a84296c6a45c4734bbe39beebb670ea.

FibDebug.java

Initial screenshot after opening file in eclipse showing a syntax error in line 3 the error message suggests missing “)”



Added closing parenthesis in line 3 and ran file returned an answer of 987 which is incorrect



The screenshot shows an IDE window titled 'FibDebug.java'. The code is as follows:

```
1 public class FibDebug{
2
3     public static void main(String[] args){
4         System.out.println(fib(15));
5     }
6
7     //The following method intends to compute the n-th Fibonacci number, but contains a bug
8     static int fib(int n) {
9         int f = 0;
10        int f0 = 1;
11        int f1 = 1;
12        while (n > 1) {
13            n--;
14            f = f0 + f1;
15            f0 = f1;
16            f1 = f; }
17        return f;
18    }
19
20 }
```

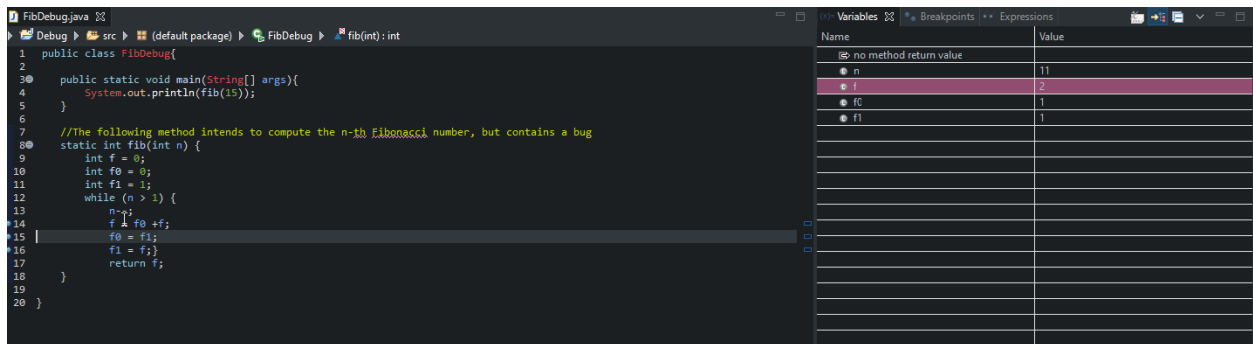
The IDE interface includes tabs for 'Console', 'Problems', and 'Debug Shell'. The 'Console' tab is active, displaying the output of the program:

```
<terminated> FibDebug [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Mar 7, 2020, 9:17:22 PM)
987
```

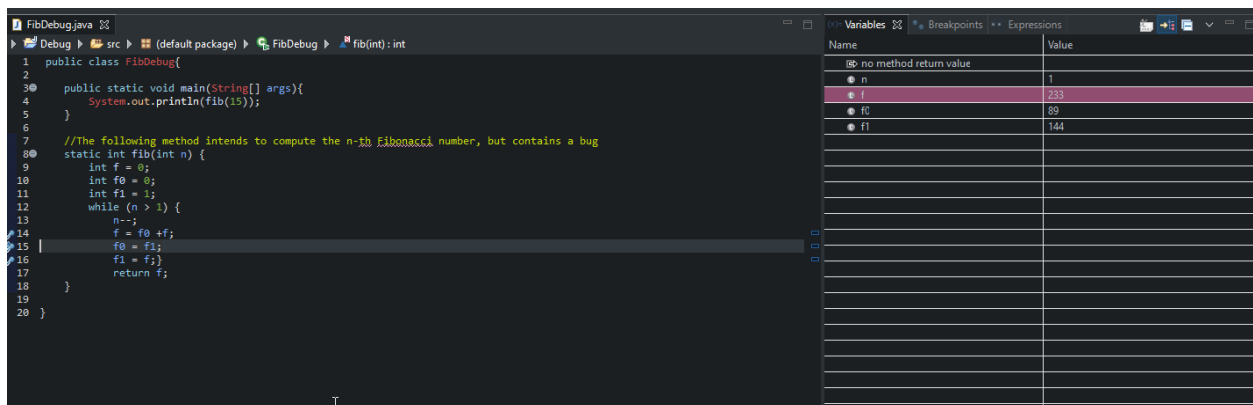
Set breakpoints at lines 14, 15, and 16 and checked the variable data and can see that the second f value is incorrect due to an incorrect starting value at line 10 for f0.



Adjusted code to get correct numbers to start the sequence. Breakpoint showing the sequence showing the correct value at the 4th position.



Breakpoint at n = 1 showing the 14th value. Now that sequence starts correctly the loop needs to run one more time and end when n = 0 not 1.



Breakpoint showing f at the proper value and with n = 0 the loop about to be broken and the 377 returned.

The screenshot shows an IDE with a Java file named `FibDebug.java`. The code is as follows:

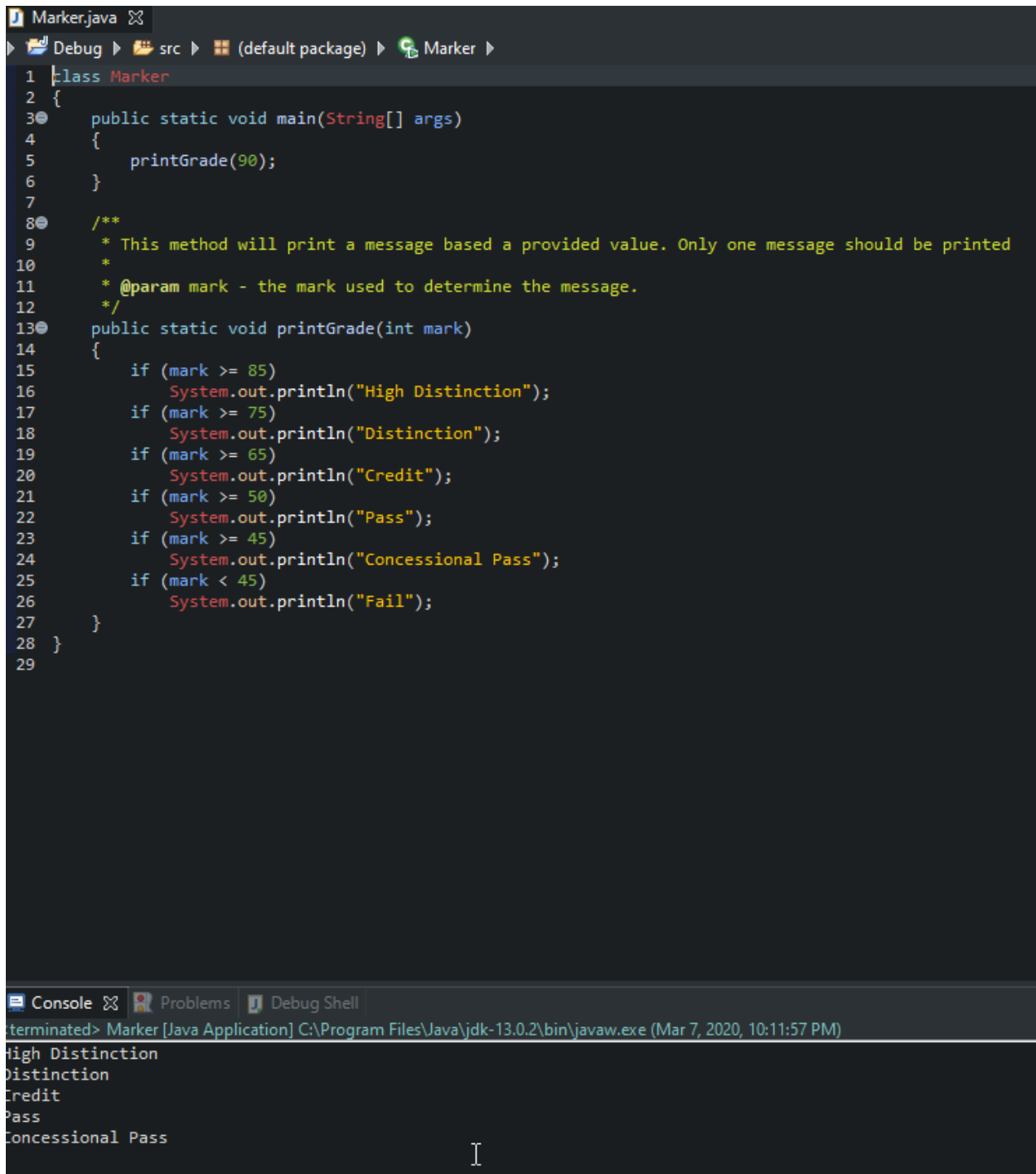
```
1 public class FibDebug{
2
3     public static void main(String[] args){
4         System.out.println(fib(15));
5     }
6
7     //The following method intends to compute the n-th Fibonacci number, but contains a bug
8     static int fib(int n) {
9         int f = 0;
10        int f0 = 0;
11        int f1 = 1;
12        while (n > 0) {
13            n--;
14            f = f0 + f1;
15            f0 = f1;
16            f1 = f;
17            return f;
18        }
19    }
20 }
```

A breakpoint is set at line 15. The `Variables` panel on the right shows the following state:

Name	Value
no method return value	
n	0
f	377
f0	144
f1	233

Marker.java

Screenshot of initial run of file showing every option being displayed when only one is desired.



The screenshot shows an IDE window titled "Marker.java" with a package explorer on the left showing the path: Debug > src > (default package) > Marker. The code in the editor is as follows:

```
1 class Marker
2 {
3     public static void main(String[] args)
4     {
5         printGrade(90);
6     }
7
8     /**
9      * This method will print a message based a provided value. Only one message should be printed
10     *
11     * @param mark - the mark used to determine the message.
12     */
13     public static void printGrade(int mark)
14     {
15         if (mark >= 85)
16             System.out.println("High Distinction");
17         if (mark >= 75)
18             System.out.println("Distinction");
19         if (mark >= 65)
20             System.out.println("Credit");
21         if (mark >= 50)
22             System.out.println("Pass");
23         if (mark >= 45)
24             System.out.println("Concessional Pass");
25         if (mark < 45)
26             System.out.println("Fail");
27     }
28 }
29
```

At the bottom, the "Console" tab is active, showing the output of the program:

```
terminated> Marker [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Mar 7, 2020, 10:11:57 PM)
High Distinction
Distinction
Credit
Pass
Concessional Pass
```

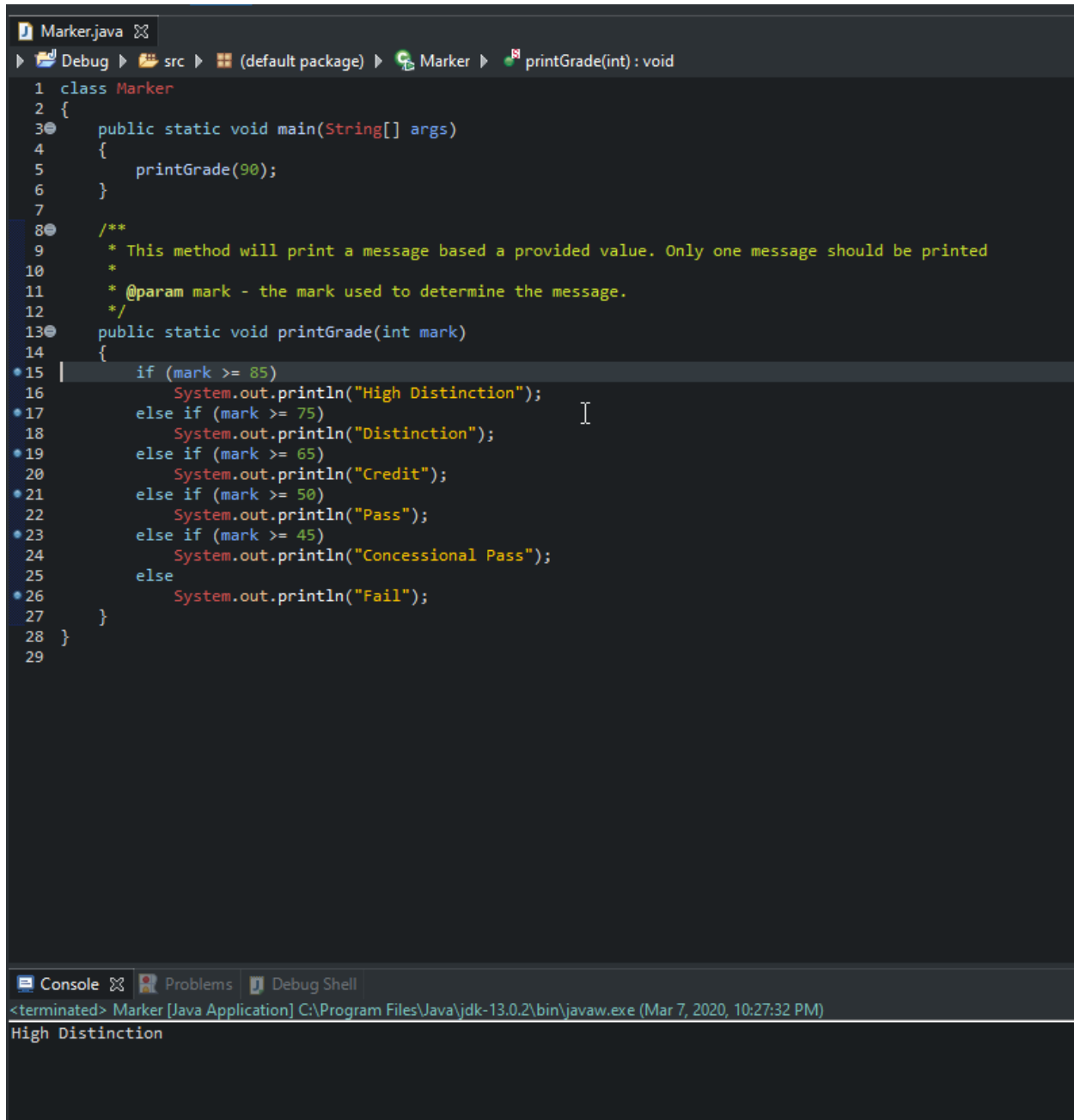
Added breakpoints at each of the if lines and it was obvious that the program currently steps through every decision point. It does not skip the others once a true value has been found.

The screenshot shows an IDE with a Java file named `Marker.java`. The code defines a `Marker` class with a `main` method and a `printGrade` method. The `main` method calls `printGrade(90)`. The `printGrade` method contains a series of if statements that print messages based on the value of `mark`. The debug console on the right shows the current state of the program, with the variable `mark` set to 90.

```
1 class Marker
2 {
3     public static void main(String[] args)
4     {
5         printGrade(90);
6     }
7
8     /**
9      * This method will print a message based a provided value. Only one message should be printed
10     *
11     * @param mark - the mark used to determine the message.
12     */
13     public static void printGrade(int mark)
14     {
15         if (mark >= 85)
16             System.out.println("High Distinction");
17         if (mark >= 75)
18             System.out.println("Distinction");
19         if (mark >= 65)
20             System.out.println("Credit");
21         if (mark >= 50)
22             System.out.println("Pass");
23         if (mark >= 45)
24             System.out.println("Concessional Pass");
25         if (mark < 45)
26             System.out.println("Fail");
27     }
28 }
29
```

Name	Value
no method return value	
mark	90

Changed if statements to else if and the final one to an else statement. Screenshot shows program having run and only breakpoint encountered was the desired one.



```
Marker.java
Debug ▸ src ▸ (default package) ▸ Marker ▸ printGrade(int) : void
1 class Marker
2 {
3     public static void main(String[] args)
4     {
5         printGrade(90);
6     }
7
8     /**
9      * This method will print a message based a provided value. Only one message should be printed
10     *
11     * @param mark - the mark used to determine the message.
12     */
13     public static void printGrade(int mark)
14     {
15         if (mark >= 85)
16             System.out.println("High Distinction");
17         else if (mark >= 75)
18             System.out.println("Distinction");
19         else if (mark >= 65)
20             System.out.println("Credit");
21         else if (mark >= 50)
22             System.out.println("Pass");
23         else if (mark >= 45)
24             System.out.println("Concessional Pass");
25         else
26             System.out.println("Fail");
27     }
28 }
29
```

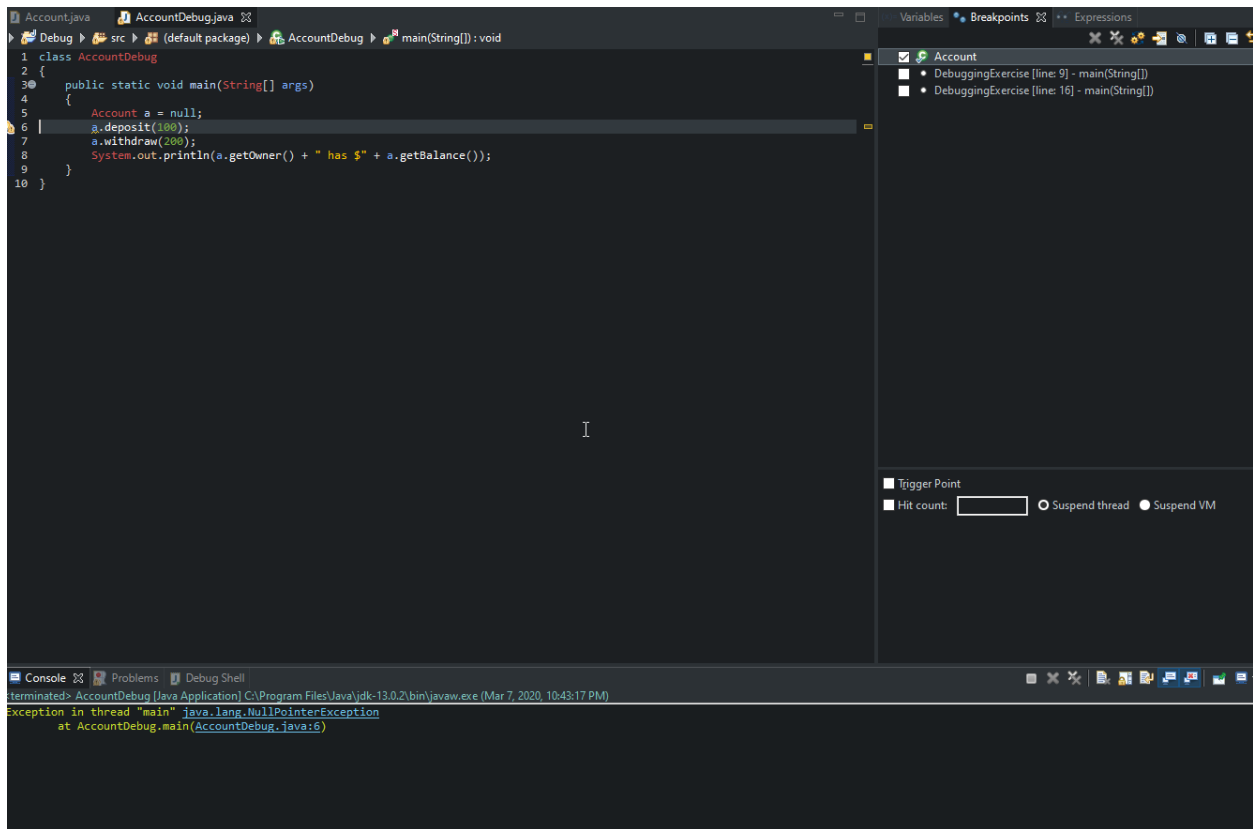
Console Problems Debug Shell

<terminated> Marker [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Mar 7, 2020, 10:27:32 PM)

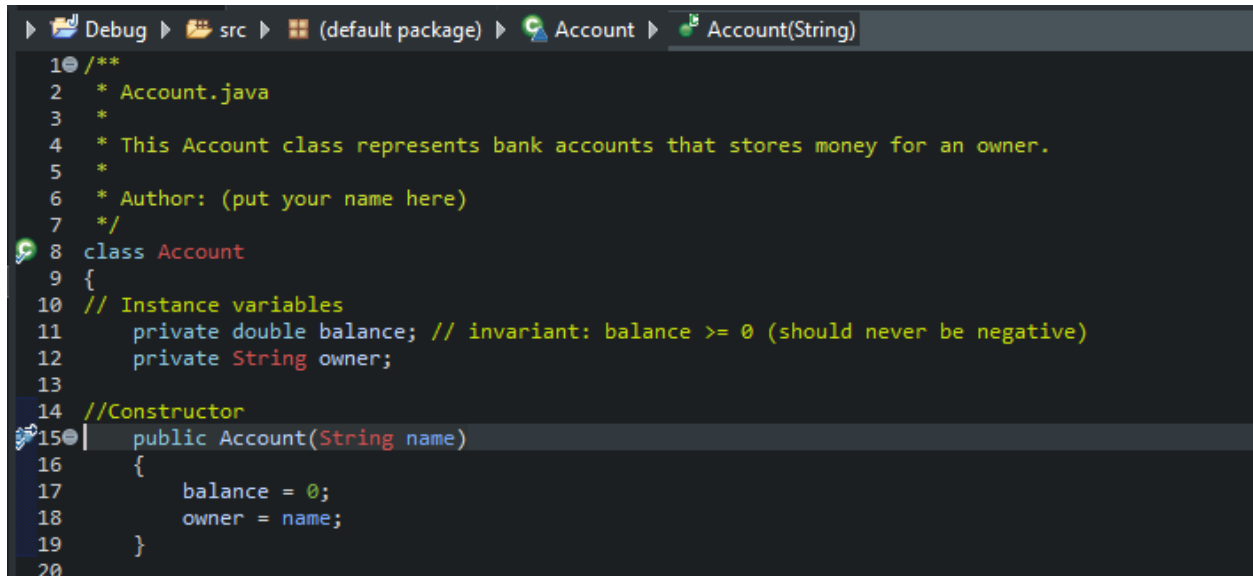
High Distinction

Account.java/AccountDebug.java

Screen shot of initial run. Showing a null pointer exception at line 6.



Changed the code in line 5 to invoke the Account object constructor properly. The following screenshot shows the debugger stopping at the breakpoint for the constructor method showing that this time it made it to that method.



The screenshot shows an IDE window with the following breadcrumb: Debug > src > (default package) > Account > Account(String). The code editor displays the following Java code:

```
1  /**
2   * Account.java
3   *
4   * This Account class represents bank accounts that stores money for an owner.
5   *
6   * Author: (put your name here)
7   */
8  class Account
9  {
10     // Instance variables
11     private double balance; // invariant: balance >= 0 (should never be negative)
12     private String owner;
13
14     //Constructor
15     public Account(String name)
16     {
17         balance = 0;
18         owner = name;
19     }
20 }
```

A green breakpoint icon is located on the left margin next to line 15, which is the start of the constructor method. The line is highlighted in a light gray background.

Screenshot of new code and successful out put in the bottom console screen:



The screenshot shows an IDE with two tabs: `Account.java` and `AccountDebug.java`. The `AccountDebug.java` tab is active, showing the following code:

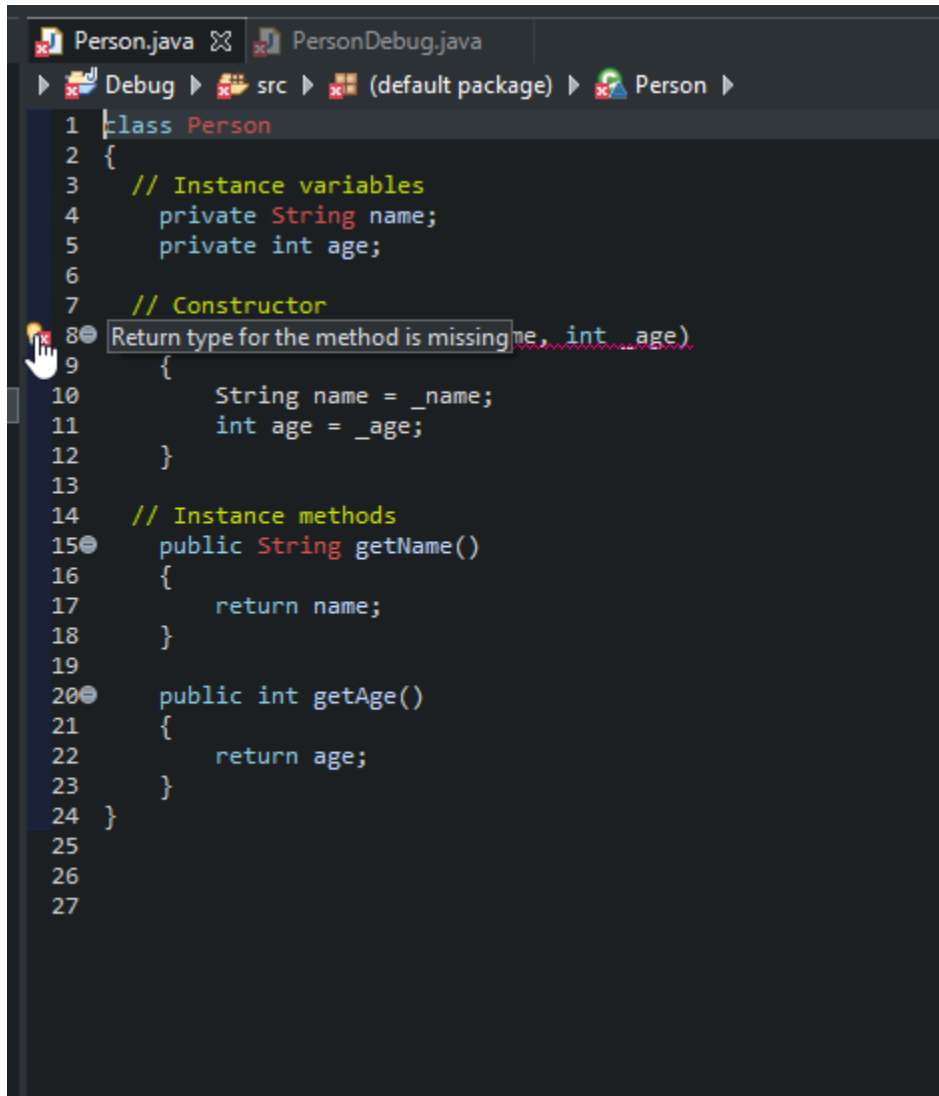
```
1 class AccountDebug
2 {
3     public static void main(String[] args)
4     {
5         Account a = new Account("Harvey");
6         a.deposit(100);
7         a.withdraw(200);
8         System.out.println(a.getOwner() + " has $" + a.getBalance());
9     }
10 }
```

The bottom of the IDE features a `Console` tab, which displays the output of the program:

```
<terminated> AccountDebug [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Mar 4
Harvey has $-100.0
```

Person.java/PersonDebug.java

Initial screenshot of the imported files in Eclipse showing the error flag and the description generated by the mouse over:



```
1 class Person
2 {
3     // Instance variables
4     private String name;
5     private int age;
6
7     // Constructor
8     Return type for the method is missing ne, int _age)
9     {
10         String name = _name;
11         int age = _age;
12     }
13
14     // Instance methods
15     public String getName()
16     {
17         return name;
18     }
19
20     public int getAge()
21     {
22         return age;
23     }
24 }
25
26
27
```

Screen shot of the PersonDebug code with the error flags and the bottom window showing the list of flagged errors:

The screenshot shows an IDE with a Java file named `PersonDebug.java`. The code is as follows:

```
1 public class PersonDebug{
2
3     public static void main(String[] args){
4         Person p = new Person("Fred Flintstone", 45);
5         System.out.println("The person is: " + p.getName() + " and is " + p.getName());
6     }
7 }
8 }
```

There are three error markers on the left side of the code editor:

- Line 3: `public static void main` - Return type for the method is missing
- Line 4: `new Person("Fred Flintstone", 45)` - Syntax error on token "statuc", static expected
- Line 4: `new Person("Fred Flintstone", 45)` - The constructor `Person(String, int)` is undefined

The bottom window shows the **Problems** tab with the following table:

Description	Resource	Path	Location	Type
Errors (3 items)				
Return type for the method is missing	Person.java	/Debug/src	line 8	Java Problem
Syntax error on token "statuc", static expected	PersonDebug...	/Debug/src	line 3	Java Problem
The constructor <code>Person(String, int)</code> is undefined	PersonDebug...	/Debug/src	line 4	Java Problem

Screenshot of the Person class. Changed the constructor method title from Student to Person to make it a constructor.

```
6
7 // Constructor
8 public Person(String _name, int _age) {
9     {
10         String name = _name;
11         int age = _age;
12     }
13 }
```

Also had to change constructor code to properly transfer data to the private variables of the object using the this parameter. As shown by the flags next to lines 10 and 11.

```
10     this.name = _name;
11     this.age = _age;
12 }
```

Screenshot of static replaced with static

```
1 public class PersonDebug{
2
3     public static void main(String[] args){
4         Person p = new Person("Fred Flintstone", 45);
5         System.out.println("The person is: " + p.getName() + " and is " + p.getName());
6     }
7
8 }
```

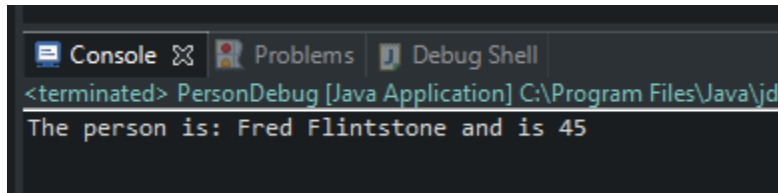
Screenshot of resulting output showing not quite what I expected:

```
Console Problems Debug Shell
<terminated> PersonDebug [Java Application] C:\Program Files\Java\jdk-13.0.2\
The person is: Fred Flintstone and is Fred Flintstone
```

A quick look at line 5 and a breakpoint at the getAge method showing it is never invoked and the second call to p.getName() is changed to p.getAge():

```
1 public class PersonDebug{
2
3     public static void main(String[] args){
4         Person p = new Person("Fred Flintstone", 45);
5         System.out.println("The person is: " + p.getName() + " and is " + p.getAge());
6     }
7
8 }
```

Desired output:



Temperature Exercise:

Screenshot of the passed unit tests for the Temperature lab:

