

DebuggingExercise.java

Original code:

```
1 class DebuggingExercise
2 {
3     public static void main(String[] args)
4     {
5         int[] numbers = new int[4];
6         for (int i=1; i<5; 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=1; i<5; i++)
15        {
16            int element = numbers[i];
17            System.out.print(element + " ");
18        }
19        System.out.println();
20    }
21 }
22
23
24
```

Bug: Got an out of bound error and would not execute.

Debugger: I used a method break point and manually went through the loop 4 times and it gave me an out of bound error. So I noticed that it was starting at index 1 instead of index 0.

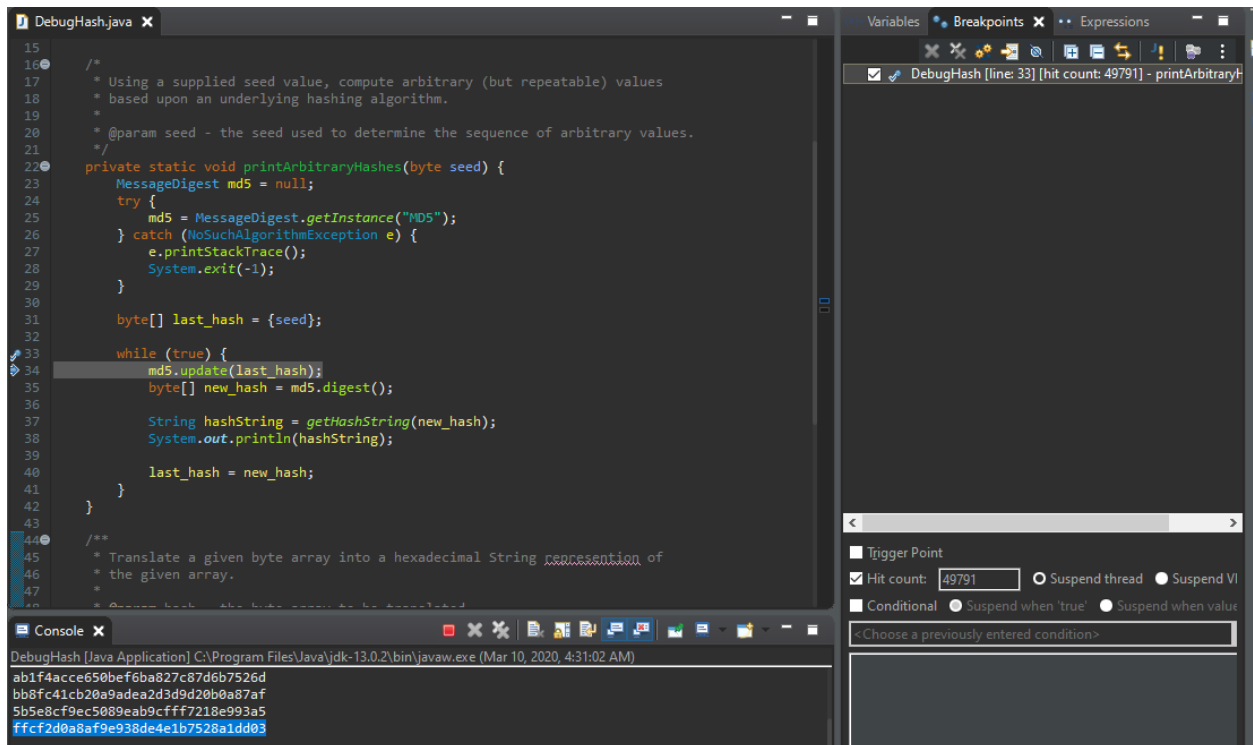
Solution: The simple solution for me was to just change the for-loop exception to < 4 from < 5 and the start value to 0, so it would iterate one less time and fill from index 0

```
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
25
```

DebugHash.java

Bug: For this one there was no bug but needed to find a specific hash number.

Debugger: I placed a break point at the while-loop where the hash was being printed and set the “Hit- Count” to 49,791 so it would halt the execution at the 49,791th hash number.



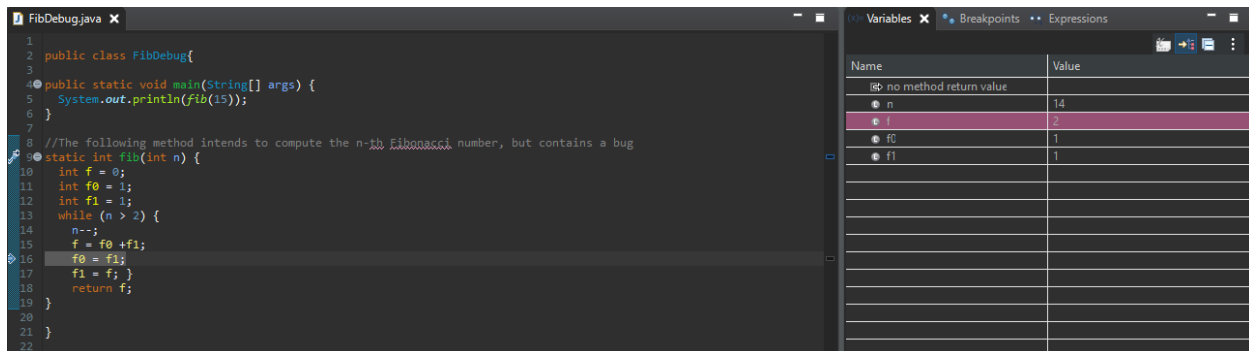
Solution: By using the “hit count” feature I was quickly able to figure out that the 49,791th hash number is: “ffcf2d0a8af9e938de4e1b7528a1dd03”

FibDebug.java

Bug: (iterating one too many times)

Debugger: I set a method break point and with the variable tab I was able to see that the value jumps to 2 from 0, skipping one step. Meaning that it will iterate one more time than it should.

Solution: The simple solution I came up with was to change while loop condition so that it would iterate one less time.



```
1 public class FibDebug{
2
3
4 public static void main(String[] args) {
5     System.out.println(fib(15));
6 }
7
8 //The following method intends to compute the n-th Fibonacci number, but contains a bug
9 static int fib(int n) {
10     int f = 0;
11     int f0 = 1;
12     int f1 = 1;
13     while (n > 2) {
14         n--;
15         f = f0 + f1;
16         f0 = f1;
17         f1 = f;
18     }
19     return f;
20 }
21
22 }
```

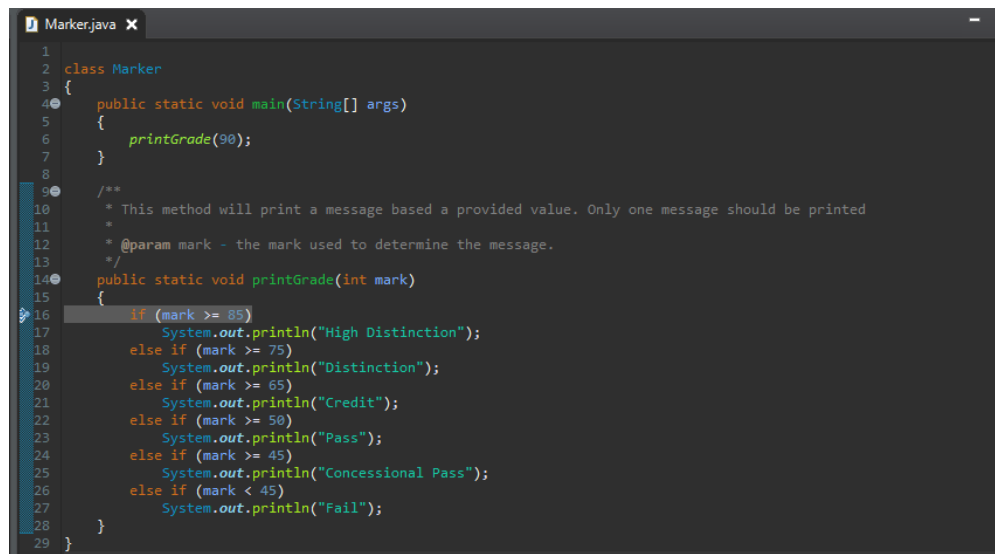
Name	Value
no method return value	
n	14
i	2
f0	1
f1	1

Maker.java

Bug: It was printing everything

Debugger: I set a break point at the beginning of the if statements. I was able to easily follow the arrow and noticed that it was testing for all the "if" statement instead of breaking out after it met its conditions.

Solution: I changed the ifs to else if statements.



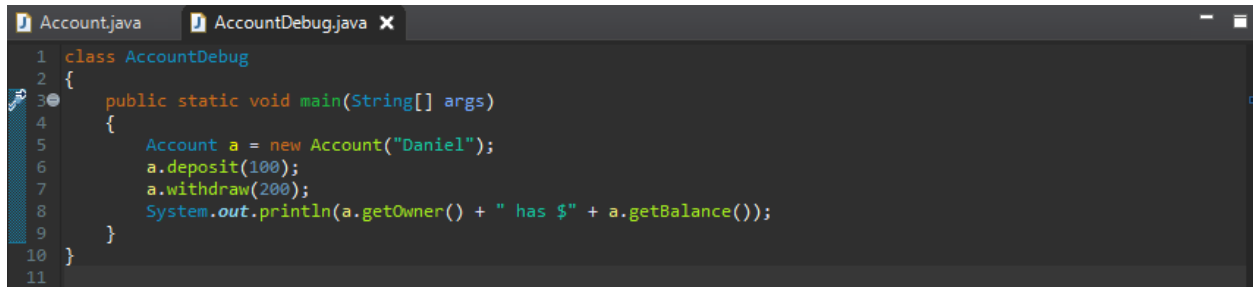
```
1
2 class Marker
3 {
4     public static void main(String[] args)
5     {
6         printGrade(90);
7     }
8
9     /**
10      * This method will print a message based a provided value. Only one message should be printed
11      *
12      * @param mark - the mark used to determine the message.
13      */
14     public static void printGrade(int mark)
15     {
16         if (mark >= 85)
17             System.out.println("High Distinction");
18         else if (mark >= 75)
19             System.out.println("Distinction");
20         else if (mark >= 65)
21             System.out.println("Credit");
22         else if (mark >= 50)
23             System.out.println("Pass");
24         else if (mark >= 45)
25             System.out.println("Concessional Pass");
26         else if (mark < 45)
27             System.out.println("Fail");
28     }
29 }
30
```

AccountDebug.java

Bug: The Account class was never initialized, so the code won't run.

Debugger: I set a method break point and ran through the code and saw that it was getting an error when it tries to deposit. So, I quickly realized that the account was never initialized.

Solution: I initialized an Account with my name.

A screenshot of an IDE window titled 'AccountDebug.java'. The code is as follows:

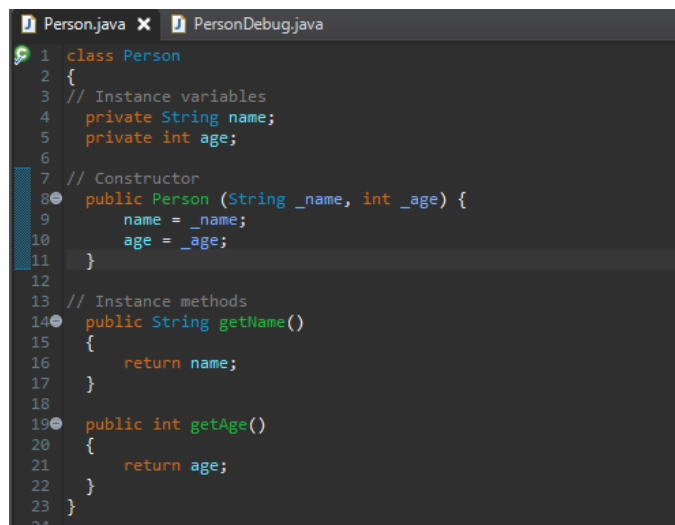
```
1 class AccountDebug
2 {
3     public static void main(String[] args)
4     {
5         Account a = new Account("Daniel");
6         a.deposit(100);
7         a.withdraw(200);
8         System.out.println(a.getOwner() + " has $" + a.getBalance());
9     }
10 }
11
```

PersonDebug.java

Bug: I don't know if the typo in the word "static" was intentional, but there is that. When I tried to run the code, it gave me an error saying that "person" constructor is undefined. After fixing that, it was not returning the correct values.

Debugger: Set a class break point. Following the debugger's steps, I knew something was up in the Person.java file when it gave me an error running line 4. Once I fixed the constructor, using the debugger's variable tab I was able to figure out that the person contractor was creating a new variable instead of storing them into the class variables.

Solution: Changed "Student" to "Person" and set the variables to store in the class variables.

A screenshot of an IDE window titled 'Person.java'. The code is as follows:

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