<u>Debugging Exercise 1:</u>

Bug Name: ArrayIndexOutOfBoundsException

```
public class ArrayManipulation
{
     public static void main(String[] args)
    {
      int[] numbers = {1, 2, 3, 4, 5};
     /* HERE THE LENGTH OF THE ARRAY IS :-  5*/
          for (int i = 0; i < numbers.length; i++)
         {
         System.out.println(numbers[i]);
        }
   }
}
```

<u>Bug of exercise -1 :</u>

_   In the Debugging Exercise 1 the bug line was for(int i=0;i<=numbers.length;i++).

The reason is as the size of the array is 5 and for accessing each array element form the array

we have to starts the index from (0 to array.length-1) .But in the Debugging Exercise 1 the array was start is from(0 to array.length) so I have debug that error.

<u>Debugging Exercise 2:</u>

<u>Bug name: cannot find symbol ,symbol : car.stop()</u>

Object-Oriented Programming Objective:

```
 class Car
{
  private String make;
 private String model;
 public Car(String make, String model)
            {
              this.make = make;
             this.model = model;
           }
public void start()
 {
   System.out.println("Starting the car.");
}
}
public class Main
  {
    public static void main(String[] args)
    {
        Car car = new Car("Toyota", "Camry");
        car.start();
   }
}
```

Bug of of exercise -2 :

As the Car class does not contain the car.stop(); method and stop() method also not defines so

It will give some error.

So I have removed that (car.stop();) part.

Debugging Exercise 3:

Bug Name ArithmeticException: / by zero

```
public class ExceptionHandling
{
     public static void main(String[] args)
     {
         int[] numbers = {1, 2, 3, 4, 5};
        try {
              System.out.println(numbers[10]);
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("Array index out of bounds.");
             }
     int result = divide(10, 0);
    System.out.println("Result: " + result);
    }
  /*Here i have handled the bug */
   public static int divide(int a, int b)
      {
       try {
           return a / b;
       }catch (Exception e)
       {
           System.out.println("We can not devide a number with zero");
       }
       return 0;


      }
}
```

__**Bug of of exercise -3**__ :

The bug in the exercise-3 f is we can not devide a number with zero and it will give ArithmeticException: / by zero exception so we have to fix that bug. So we can fix this type of exception by using try and catch block in our code.

So I wrote the risky code in the try and catch block.

__**Exercise 4:**__

Bug :- Position Mismatch as required form user

```
 public class Fibonacci {
       public static int fibonacci(int n)
       {
            if (n <= 1)
                  return n;
           else
                 return fibonacci(n-1) + fibonacci(n-2);
     }
     public static void main(String[] args)
    {
        int n = 6;

       int result = fibonacci(n-1);

     System.out.println("The Fibonacci number at position " + n + " is: " + result);

   }

 }
```

Bug of of exercise -4 :

- ➔ The bug in the exercise-4 , The Fibonacci series is 0 ,1, 1, 2, 3,5,8,11,19……..

So if according to the exercise-4 code if a user want to see the Fibonacci number form the user entered position the code will shows the Fibonacci number present at (entered position +1)th location .

- ➔ according to the code is a user want to know the exact Fibonacci number the we have to write

as fibonacci(n-1) where n is the entered position.

So hava have changed the   code line form fibonacci(n) to fibonacci(n-1).

- ➔ Now the code will return the exact Fibonacci number according to the specified n (position value).

And specified base condition is currect.

Exercise 5:

Bug: illegal start of type

```
import java.util.*;


public class PrimeNumbers {

   public static List<Integer> findPrimes(int n) {

      List<Integer> primes = new ArrayList<Integer>();


      for (int i = 2; i <= n; i++) {

         boolean isPrime = true;


         for (int j = 2; j <= Math.sqrt(i); j++) {

            if (i % j == 0) {

               isPrime = false;
```

```
                break;
            }
        }

        if (isPrime) {
            primes.add(i);
        }
    }

    return primes;
}

public static void main(String[] args) {
    int n = 20;
    List<Integer> primeNumbers = findPrimes(n);
    System.out.println("Prime numbers up to " + n + ": " + primeNumbers);
}
}
```

<u>Bug of of exercise -5</u> :

I have change the line 4 as List<Integer> primes = new ArrayList<Integer>(); in the constructor because it is showing an error and in the logic for calculating a prime number we can change the condition in the inner for loop ,Although the logic is also runs the program smoothly but using that

(j<i) logic we can change as j<Math.sqrt(i). so the time complexity will be efficient.

Submitted by,

Priyanshu Meher

Java Programming Intern(5th Jan Batch)

Gmail:- sibupriyanshu292@gmail.com

Dt:-08th Jan 2024