

Task 1:

Debugging Exercise 1: Array Manipulation

1. Error: Array out of bound

2. Code:

```
public class ArrayManipulation {  
    public static void main(String[] args) {  
        int[] numbers = { 1, 2, 3, 4, 5 };  
        for (int i = 0; i < numbers.length; i++) {  
            System.out.println(numbers[i]);  
        }  
    }  
}
```

3. **Error Explanation:** We have to reduce the reptation of for loop. There is `<=numbers.length` to `<numbers.length`.

Debugging Exercise 2: Object-Oriented Programming

1. **Error:** No declaration of method stop

2. **Code :**

```
class Car {  
    private String make;  
    private String model;  
  
    public Car(String make, String model) {  
        this.make = make;  
        this.model = model;  
        System.out.println("This car is made by "+this.make+" and mode is "+this.model);  
    }  
  
    public void start() {  
        System.out.println("Starting the car.");  
    }  
    //Declaring the method stop  
    public void stop(){  
        System.out.println("Stoping the car"); //Error sloved  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Car car = new Car("Toyota", "Camry");  
        car.start();  
        car.stop(); //We have error in this line, We have to declare the method
```

3. **Error Explanation:** In the main method you call a method but it was not declared. So we are facing error. To slove that I decalred a method call stop in the car class.

Debugging Exercise 3: Exception Handling

1.Error: Arithmetic Exception in method divide.

2. Code:

```
package Valuteofcodes;
```

```
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);  
    }  
  
    public static int divide(int a, int b) {  
        try {return a / b; } //Exception occurs here  
        catch(ArithmeticException e){  
            System.out.println("Enter the correct value of b ");  
            return -1;  
        }  
    }  
}
```

3.Error Explanation: When it divide a by b. It is dividing by 0 so, we will get run time exception that is Arithmetic Exception. I solved it by using try-catch exception method.

Exercise 4:

1. **Error:** Exceeding n value

2. **Code :**

```
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);  
}  
}
```

3. **Error Explanation:** If we pass Fibonacci(n) it is displaying n+1th value, so we reducing value to (n-1) .

Exercise 5:

1. **Error:** *(Reptation of 2nd for loop)

2. **Code :**

```
import java.util.*;
```

```
public class PrimeNumbers {
    public static List<Integer> findPrimes(int n) {
        List<Integer> primes = new ArrayList<>();
        for (int i = 2; i <= n; i++) {
            boolean isPrime = true;
            for (int j = 2; j < i/2; 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);
    }
}
```

3. **Error Explanation:** We can check the j upto $i/2$. No need to check up to $i-1$.