**Assignment 2**

**Debugging Exercise 1: Array Manipulation**

Objective: To identify and fix errors in a Java program that manipulates arrays.

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]);

        }

    }

}

In the above code we get a Runtime Exception saying: **java.lang.ArrayIndexOutOfBoundsException** at condition in the for loop.

Reason: We get such Exception when we are trying to access the index of array which is greater than size of the array.

**for (int i = 0; i <= numbers.length; i++)**

Here size of array is 5. But we are trying to access the 6th element in the above line.

**We can resolve this by using any of the following**

1. **for (int i = 0; i < numbers.length; i++)** //by replacing **<=** operator to **<** this repeats loop till last elemnt in the array

OR

1. **for (int i = 0; i <= numbers.length-1; i++)**

//by replacing **numbers.length** with **numbers.length-1**  this repeats loop till last element in the array

OR

1. **for (int i = 1; i <= numbers.length-1; i++)**  //or by simply starting the index with **1** we can resolve.

Output before debugging:

1

2

3

4

5

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

at ArrayManipulation.main(Test.java:6)

The debugged 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]);

        }

    }

}

The output after debugging:

1

2

3

4

5

**Debugging Exercise 2: Object-Oriented Programming**

Objective: To identify and fix errors in a Java program that demonstrates basic object-oriented programming principles.  
  
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();

        car.stop();

    }

}

In the above code we get a compile time error saying:

Test.java:20: error: cannot find symbol

**car.stop();**

symbol: method stop()

location: variable car of type Car

1 error

Reason : we don’t have a method named stop in the class named **Car**

We resolve this by simply

1. remove Car.stop() method call in main() method

OR

1. declare a method stop() in the Car class

We can create a method stop() like

public void stop(){

        System.out.println("Stoping the car.");

    }

………………………………………………………………………………………

**Debugging Exercise 3: Exception Handling**

Objective: To identify and fix errors in a Java program that demonstrates exception handling.

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) {

        return a / b;

    }

}

In the above code there are multiple errors

1. we are trying to divide a number with zero which causes a Runtime Exception and that Exception is not handled.

We can resolve this by placing the exception causing lines in the try block and corresponding catch block .

The solution to overcome this is:

Replace the bold highlighted lines in the above code with the below lines

try{

            int result = divide(10, 0);

            System.out.println("Result: " + result);

        } catch(ArithmeticException e){

            System.out.println("Arithmetic Exception: / by zero");

        }

**Exercise 4:**  
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);

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

    }

}

The above code doesn’t contain any bugs. It has no errors and providing correct output.

…………………………………………………………………

**Exercise5**:

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; 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);

    }

}

The above code doesn’t contain any bugs. It has no errors and providing correct output.