VaultOfCodes

Assignment 1

Task 1

Review the following codes, Find and fix errors also explain the errors

Debugging Exercise 1:Array Manipulation

Objective:To identify and fix error in a java program that manipulation 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]);

        }

    }

    }
```

Correct 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]);

}}}
```

Explain error: /* To fix this, we should change the loop condition to `i < numbers.length */`

Debugging Exercise 2:Object -Oriented Programming

Objective: To Identify and fix error 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();
```

```
  }}
```

Correct code:

```
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 void stop() {
        System.out.println("Stopping the car.");
    }
}

public class Main {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Camry");
        car.start();

        car.stop();

    }}
```

Error explain:

/* Changed the class name `main` to `Main`.

Added the `stop` method to the `Car` class./**

Debugging Exercise 3: Exception Handling**

Objective: To identify asnd fix error in a 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;

    }

    }
```

Correct code:

```
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");

        }


        try {

            int result = divide(10, 0);

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

        } catch (ArithmeticException e) {

            System.out.println("Cannot divide by zero");
        }
    }

    public static int divide(int a, int b) {
        return a / b;
    }
}
```

Error explain:

/* The division by zero error is not handled, which will cause an ArithmeticException*/

/* • Added a try-catch block around the call to divide in the main method to catch an ArithmeticException for division by zero.

• Kept the original try-catch block to handle the ArrayIndexOutOfBoundsException./**

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 code aims to calculate the fibonacci sequence .However there is bug in the code .When the student runs this code, it will raise an error or product incorrect output.The student task is to identify and correct the bug

Correct 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);

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

```
    }
}
```

Explain error:

The provided code has no syntax errors and will compile and run correctly

Excersise 5:

```
import java.util.*;
public class PrimeNumber{
    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 number up to " + n + " : " + primeNumbers );
    }
}
```

The code aims to find prime number up to a given limit .However there is bug in the code .

When the student runs this code, it will raise an error or product incorrect output.The student task is to identify and correct the bug

Hint: Check he condition for checking prime number

Correct code:

```
import java.util.*;

public class PrimeNumber {
    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 <= 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);
    }
}
```

Explain error:

Optimized the inner loop to check divisors only up to the square root of `i`

```
[       for (int j = 2; j <= Math.sqrt(i); j++) ]
```