1. **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]);**

**}**

**}**

**}**

**Ans:-**

Here we are getting the ArrayIndexOutOfBoundsException from the questions code.

Because we are having only five elements in the Array, but in the for loop, it is asking 6 elements. There is not having 6th Element in the Given Array , if we just remove that equal symbol in the condition of for loop , then this code will be Executed .

i.e.

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]);

}

}

}

**Output After Correction :**

1

2

3

4

5

1. **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();**

**}**

**}**

**Ans:-**

Here , we can see that in the main method , we are calling start and stop method. Where stop method does not exist . So , we have two options here . We can create a new method stop or we can remove car.stop() . If we do any one of the following then we can execute this program easily.

i.e.

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();

}

}

**OR**

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();

}

}

**Output After Correction :**

Starting the car.

Stopping the car.

**OR**

Starting the car.

1. **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;**

**}**

**}**

**Ans :-**

Here we can see that we are using Exception handling method at the main method. i.e. try catch Exception . But inside main , we are calling divide method which is holding values 10,0. We know that anything divide by Zero is Infinity. Here 10 is divided by 0 ,which means we must use exception inside the divide method .Because it lacks Exception handling, Default exception will handle this . Here we are adding a try catch Exception in it

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;

}

catch(ArithmeticException ae) {

System.***out***.println("Divide by Zero , Arithmetic Exception ");

}

return -1;

}

**Output After Correction :**

Array index out of bounds.

Divide by Zero , Arithmetic Exception

Result: -1

1. **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 a bug in the code. When the student runs this code, it will raise an error or produce incorrect output. The student's task is to identify and correct the bug.**

**Hint: Pay close attention to the base case and recursive calls.**

**Ans :-**

In this code , we are trying to get the Fibonacci number at the given position . The given above is a Correct one , If we use this we can get Fibonacci number at given position . This code is Not having any Exception or Errors.

**Output :**

The Fibonacci number at position 6 is: 8

1. **Exercise 5:  
   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 code aims to find prime numbers up to a given limit. However, there is a bug in the code. When the student runs this code, it will raise an error or produce incorrect output. The student's task is to identify and correct the bug.**

**Hint: Check the condition for checking prime numbers.**

**Ans:-**

In this code , We are trying to find whether the given number is prime or not. The above code is correct as before . This Code is Giving us all Prime Number until given number .

**Output:**

Prime numbers up to 20: [2, 3, 5, 7, 11, 13, 17, 19]