Exercise 1 :

Error code :

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

corrected code :

public class ArrayManipulation {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

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

System.out.println(numbers[i]);

}

}

}

Explaination of the error:

The condition i <= numbers.length in the loop will cause an ArrayIndexOutOfBoundsException because array indices are zero-based, and the loop condition should be i < numbers.length to access the elements within the valid index range (0 to numbers.length - 1).

Exercise 2 :

Error code :

car.stop();

Corrected code :

we can remove the stop() method because in main(car) method does not contain stop()method.

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();

    }

}

Explaination of the error:

The result is compilation error because the Car class does not have a method called stop() defined within it. The Main class is trying to call car.stop(), which the Car class does not support.

Exercise 3 :

Error code :

int result = divide(10, 0)

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

}

public static int divide(int a, int b) {

return a / b;

}

}

Corrected 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) {

try {

return a / b;

} catch (ArithmeticException e) {

throw new ArithmeticException("Cannot divide by zero.");

}

}

}

Explaination of the error:

Here ArithmeticException.

The issue with this code is that it attempts division by zero in the divide method without explicitly handling the potential ArithmeticException. To fix this, you should incorporate a try-catch block to handle the potential division by zero .

Exercise 4:

Error code :

for (int j = 2; j < i; j++)

Corrected 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 \* j <= i; j++) { // Changed the condition here

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);

}

}

Explaination of the error:

In the given code the occurs in inner for() loop . The code checks if a number is prime by testing if it's divisible by any number smaller than itself. However, it incorrectly marks certain numbers as prime that aren't.

