

# ASSIGNMENT-7.3

2303a51595

B-10

## TASK-1 : Fixing Syntax Errors

### Prompt :

Write a Java method add(int a, int b) that returns the sum of two numbers. Detect and correct any syntax errors in the method definition.

### Wrong Code (with syntax error)

```
int add(int a, int b) // missing {  
    return a + b;
```

### Code :

```
public class AddFunction {  
    static int add(int a, int b) {  
        return a + b;  
    }  
    public static void main(String[] args) {  
        System.out.println(add(5, 3));  
    }  
}
```

### Output :

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows multiple Java files including `AddFunction.java`, `duplicate.java`, `PrimeNum.java`, `Student.java`, `StudentTest.java`, and `AddFunction.java` (the current file).
- Code Editor:** Displays the `AddFunction.java` code with syntax highlighting and a cursor at the end of the first line.
- Terminal:** Shows the command-line interface with the following logs:
  - Java version: Microsoft Windows [Version 10.0.26200.7705]
  - Copyright notice: (c) Microsoft Corporation. All rights reserved.
  - Execution command: C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac AddFunction.java
  - Execution output: C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java AddFunction
  - Output: 8
  - Final path: C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE\]
- Status Bar:** Shows the current line (Ln 6), column (Col 1), spaces (Spaces: 4), encoding (UTF-8), line endings (CRLF), Java language level ({}), and Go Live status.

## Analysis :

The error was caused by a missing opening { after the method definition. After adding {}, the method works correctly and returns the sum of two numbers.

## TASK-2 : Debugging Logic Errors in Loops

### Prompt :

Write a Java program with a countdown loop from n to 0. Detect and fix any increment/decrement mistake that may cause an infinite loop.

### Wrong Code (with logic error)

```
int n = 5;
while (n >= 0) {
    System.out.println(n);
    n++; // wrong (causes infinite loop)
}
```

### Code :

```
public class CountDown {
    public static void main(String[] args) {
        int n = 5;
        while (n >= 0) {
            System.out.println(n);
            n--; // correct decrement
        }
    }
}
```

### Output :

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows various Java files and classes, including .vscode, PrimeNum.java, Student.java, StudentTest.java, AddFunction.java, and CountDown.java.
- Code Editor:** Displays the CountDown.java code. The line `n++;` is highlighted in red, indicating an error.
- Terminal:** Shows the command-line interface where the code is being run. It includes:
  - `C:\Users\deepthi\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac AddFunction.java`
  - `C:\Users\deepthi\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java AddFunction`
  - `8`
  - `C:\Users\deepthi\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac CountDown.java`
  - `C:\Users\deepthi\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java CountDown`
  - `5`
  - `4`
  - `3`
  - `2`
  - `1`
  - `0`
- Status Bar:** Shows the current file is CountDown.java, with line 7, column 10, and other standard status information.

## Analysis :

The loop used n++, which increases the value and never stops, causing an infinite loop.

Changing it to n-- decreases the value each time and stops the loop correctly at 0.

## TASK-3 : Debugging Logic Errors in Loops

### Prompt :

Write a Java program to create a function that performs division of two numbers and handles runtime errors using try-cat blocks.

### Code :

```
class DivideDemo {  
    static int divide(int a, int b) {  
        return a / b;  
    }  
    public static void main(String[] args) {  
        try {  
            System.out.println(divide(10, 0));  
        }  
        catch (ArithmaticException e) {  
            System.out.println("Error: Cannot divide by zero!");  
        }  
    }  
}
```

### Output :

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree with several Java files, including DivideDemo.java, which is currently selected. The main editor area shows the provided Java code. Below the editor, the terminal tab is active, displaying the command-line output of running the Java program:

```
C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac DivideDemo.java  
C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java DivideDemo  
Error: Cannot divide by zero!
```

The status bar at the bottom indicates the current file is main\*, and the terminal shows three cmd instances running.

### **Analysis :**

Dividing a number by zero causes an `ArithmaticException` at runtime and crashes the program.

Using try-catch handles the error safely and prints a message instead of stopping execution.

### **TASK-4 : Debugging Class Definition Errors**

#### **Prompt :**

Create a Java class `Rectangle` with length and width. The constructor is defined incorrectly. Identify the error and fix the class.

#### **Wrong Code (with errors)**

```
class Rectangle {  
    int length;  
    int width;  
    void Rectangle(int l, int w) { // X wrong: constructor has  
    return type  
        length = l;  
        width = w;  
    }
```

#### **CODE :**

```
class Rectangle {  
    int length;  
    int width;  
    // ✓ correct constructor (no return type)  
    Rectangle(int l, int w) {  
        length = l;  
        width = w;  
    }  
    void display() {  
        System.out.println("Length: " + length);  
        System.out.println("Width: " + width);  
    }  
}
```

```

public static void main(String[] args) {
    Rectangle r = new Rectangle(10, 5);
    r.display();
}

```

## Output :

```

File Edit Selection View Go ...
File Explorer AI ASSISTED CODE
duplicate.java J NameFormatter.java U J VowelCounterZeroShot.java U J java U J DivideDemo.java U J Rectangle.java U ...
RECTANGLE.java
1  class Rectangle {
2
3      int length;
4      int width;
5
6      // checked correct constructor (no return type)
7      Rectangle(int l, int w) {
8          length = l;
9          width = w;
10     }
11
12     void display() {
13         System.out.println("Length: " + length);
14         System.out.println("Width: " + width);
15     }
16
17     public static void main(String[] args) {
18         Rectangle r = new Rectangle(10, 5);
19         r.display();
20     }
21 }

```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL GITLENS

C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java Rectangle  
Length: 10  
Width: 5

Ln 1, Col 7 (10 selected) Spaces: 4 UTF-8 CRLF {} Java Go Live

## Analysis :

The constructor wrongly had a return type void, so Java treated it like a normal method. Removing the return type makes it a proper constructor that initializes the object correctly.

## TASK-5 : Resolving Index Errors in Lists

### Prompt :

Create an array and try to access an invalid index.

Identify the runtime error and fix it using exception handling or bounds checking.

```

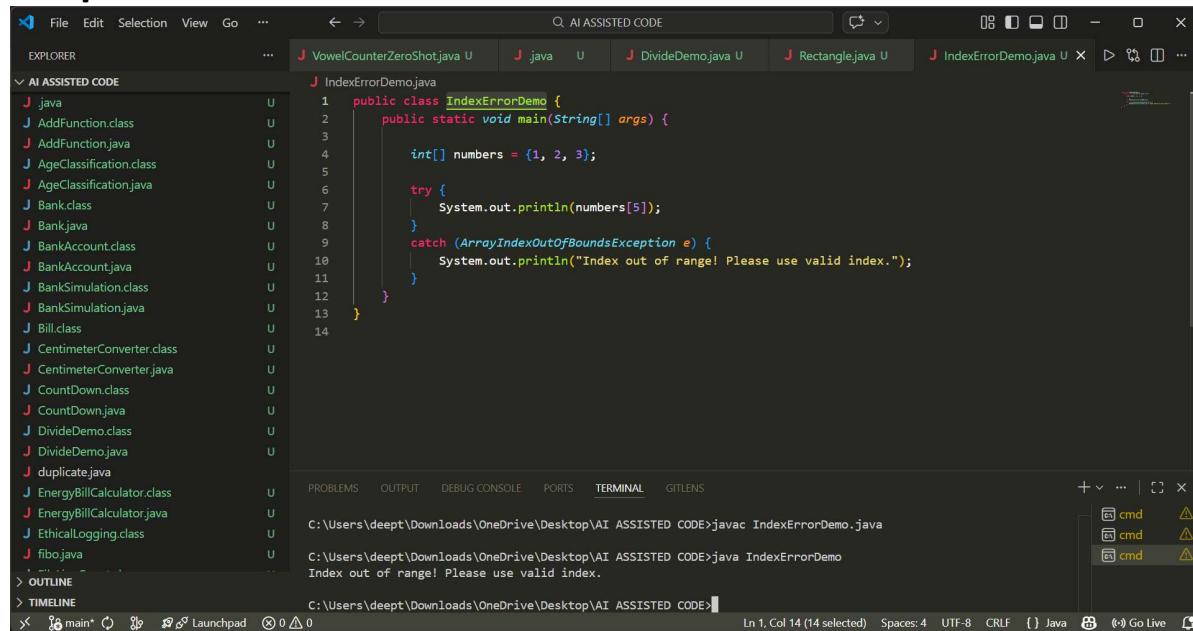
public class IndexErrorDemo {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3};
        // ✗ invalid index
        System.out.println(numbers[5]);
    }
}

```

## Code :

```
public class IndexErrorDemo {  
    public static void main(String[] args) {  
        int[] numbers = {1, 2, 3};  
        try {  
            System.out.println(numbers[5]);  
        }  
        catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Index out of range! Please use valid  
index.");  
        }  
    }  
}
```

## Output :



The screenshot shows a Java development environment with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Toolbar:** Back, Forward, Search, Refresh, Minimize, Maximize, Close.
- Left Sidebar:** EXPLORER (shows various Java files like VowelCounterZeroShot.java, DivideDemo.java, Rectangle.java, etc.) and AI ASSISTED CODE (selected tab, showing the IndexErrorDemo.java code).
- Code Editor:** Displays the IndexErrorDemo.java code with line numbers 1 to 14.
- Terminal:** Shows the command line output:
  - C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac IndexErrorDemo.java
  - C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java IndexErrorDemo
  - Index out of range! Please use valid index.
- Status Bar:** Ln 1, Col 14 (14 selected), Spaces: 4, UTF-8, CRLF, {}, Java, Go Live.

## Analysis :

The program crashes because we accessed an index outside the array size.

Using try-catch or checking bounds prevents the program from crashing and handles the error safely.