

A. Informasi & Ringkasan**1. Identitas**

NIM	11S23043
Nama	Grace Evelin Siallagan
Kelas	12 IF 2
Judul Praktikum	Review PBO
Video Presentasi	https://youtu.be/crXRSRxuf1k?si=6T0rf75Xh1ue_w1u

2. Capaian & Ringkasan Pemahaman

Hasil Capaian setelah mengikuti praktikum:

- A (Penyelesaian): Selesai (1), Tidak Selesai (0).
- B (Pemahaman): Tidak Paham (1), Kurang Paham (2), Cukup Paham (3), Paham (4), Sangat Paham (5).

No	Indikator	A	B
1	Belajar Pemrograman Java	1	3
2	Studi Kasus 1: Matrix App	1	3

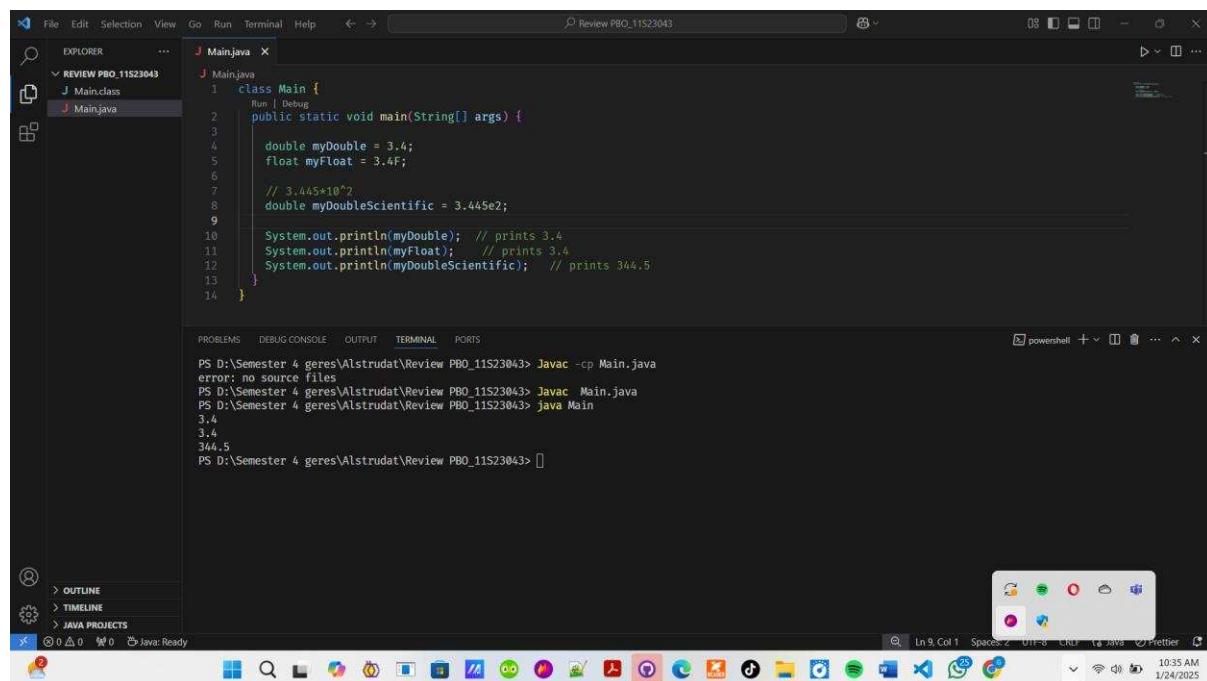
Dalam pemahaman saya, setelah mempelajari dan mempraktikkan praktikum ini, mahasiswa diharapkan mampu memahami kembali konsep pemrograman berorientasi objek (PBO) secara mendalam, khususnya dalam konteks pemrograman Java. Mahasiswa juga diharapkan dapat menerapkan prinsip-prinsip PBO seperti enkapsulasi, pewarisan, dan polimorfisme untuk menyelesaikan studi kasus nyata, seperti pengelolaan matriks yang mencakup operasi penjumlahan, pengurangan, dan perkalian. Praktikum ini melatih kemampuan teknis dalam coding, analisis logika, serta pengembangan program modular menggunakan file terpisah, sehingga mendukung pembelajaran yang lebih aplikatif dan relevan dengan kebutuhan industri pemrograman.

B. Laporan Aktivitas Praktikum

1. Belajar Pemrograman Java

• Java Fundamental

Floating-point Literals



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

- File Structure:** The left sidebar shows a project named "REVIEW PBO_11S23043" containing a file "Main.java".
- Code Editor:** The main area displays the following Java code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         double myDouble = 3.4;  
4         float myFloat = 3.4F;  
5         // 3.445*10^2.  
6         double myDoubleScientific = 3.445e2;  
7         System.out.println(myDouble); // prints 3.4  
8         System.out.println(myFloat); // prints 3.4  
9         System.out.println(myDoubleScientific); // prints 344.5  
10    }  
11 }
```
- Terminal:** Below the code editor, the terminal window shows the command-line output of running the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac -cp Main.java  
error: no source files  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
3.4  
3.4  
344.5  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- System Tray:** The bottom right corner shows the Windows system tray with icons for battery, signal, and date/time.

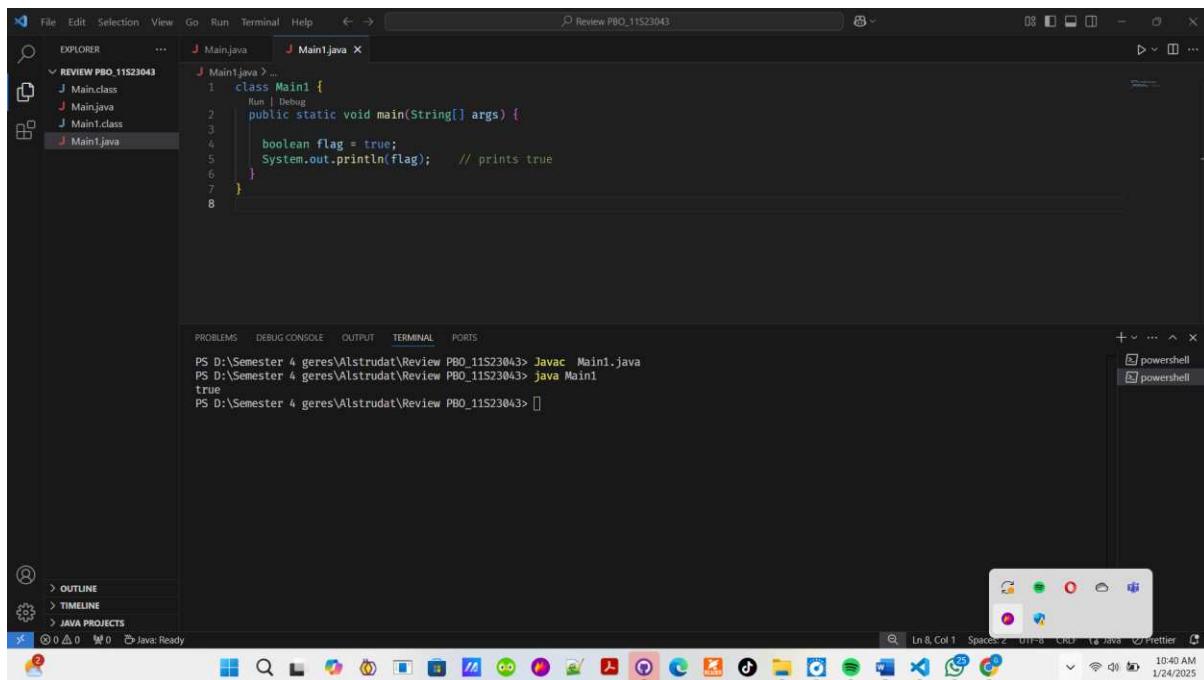
Kode tersebut mendeklarasikan tiga variabel untuk menyimpan bilangan desimal. Variabel myDouble bertipe double dengan nilai **3.4**, sementara myFloat bertipe float dengan nilai **3.4F**. Variabel myDoubleScientific menggunakan notasi ilmiah **3.445e2**, yang berarti **3.445×10^2** atau **344.5**.

✓ Java Data Types (Primitive)

Java boolean data type

11S2215 - Algorithms and Data Structures

Laporan Praktikum



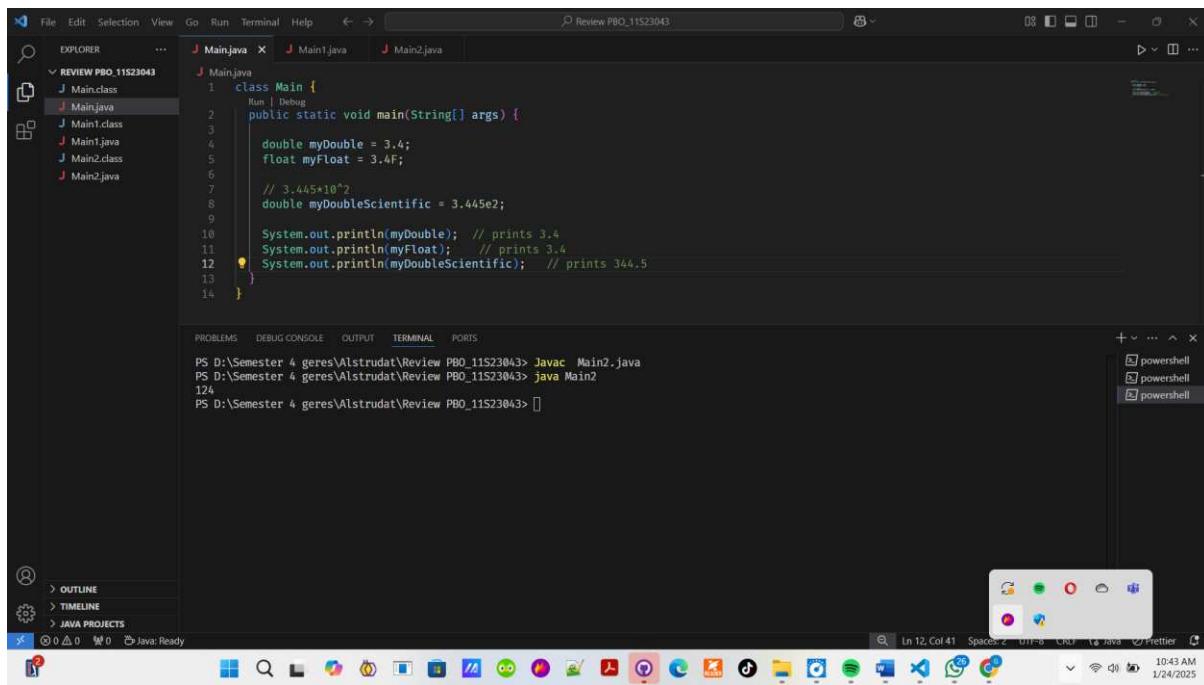
A screenshot of the Visual Studio Code (VS Code) interface. The top bar shows the menu: File, Edit, Selection, View, Go, Run, Terminal, Help. The title bar says "Review PBO_11S23043". The Explorer sidebar on the left shows a project named "REVIEW PBO_11S23043" containing files Main.class, Main.java, Main1.class, and Main1.java. The main editor window displays the following Java code:

```
class Main {  
    public static void main(String[] args) {  
        boolean flag = true;  
        System.out.println(flag); // prints true  
    }  
}
```

The terminal at the bottom shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main1.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main1  
true  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java byte data type



A screenshot of the Visual Studio Code (VS Code) interface. The top bar shows the menu: File, Edit, Selection, View, Go, Run, Terminal, Help. The title bar says "Review PBO_11S23043". The Explorer sidebar on the left shows a project named "REVIEW PBO_11S23043" containing files Main.class, Main.java, Main1.class, Main1.java, Main2.class, and Main2.java. The main editor window displays the following Java code:

```
class Main {  
    public static void main(String[] args) {  
        double myDouble = 3.4;  
        float myFloat = 3.4F;  
        // 3.445*10^-2  
        double myDoubleScientific = 3.445e2;  
        System.out.println(myDouble); // prints 3.4  
        System.out.println(myFloat); // prints 3.4  
        System.out.println(myDoubleScientific); // prints 344.5  
    }  
}
```

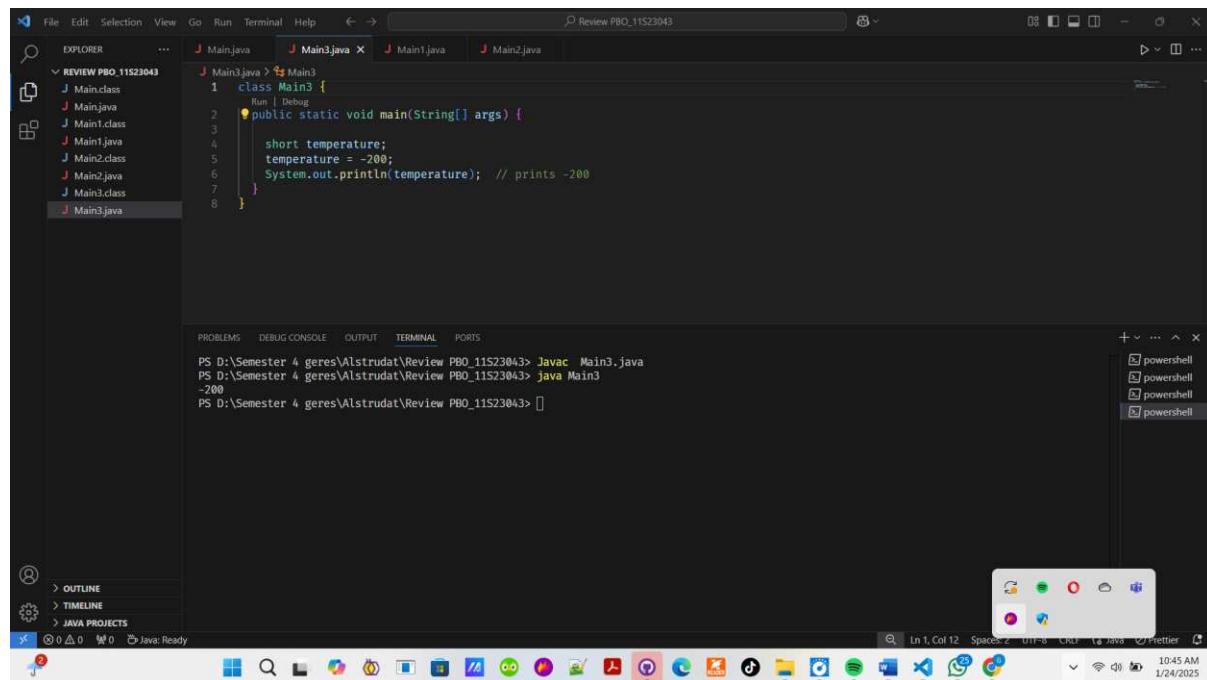
The terminal at the bottom shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main2.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main2  
124  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java short data type

11S2215 - Algorithms and Data Structures

Laporan Praktikum



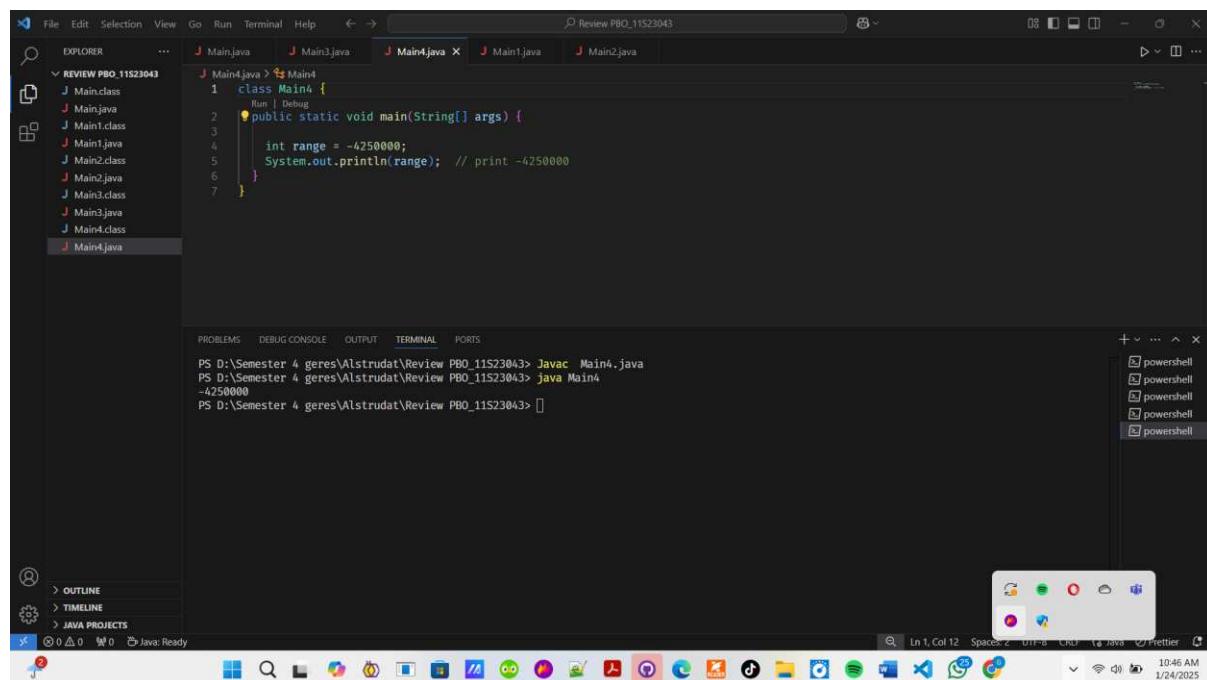
The screenshot shows a Java application running in a terminal window. The code in Main3.java is:

```
1 class Main3 {
2     public static void main(String[] args) {
3         short temperature;
4         temperature = -200;
5         System.out.println(temperature); // prints -200
6     }
7 }
```

The terminal output shows the execution of the Java command and the resulting output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main3
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main3
-200
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java int data type



The screenshot shows a Java application running in a terminal window. The code in Main4.java is:

```
1 class Main4 {
2     public static void main(String[] args) {
3         int range = -4250000;
4         System.out.println(range); // print -4250000
5     }
6 }
```

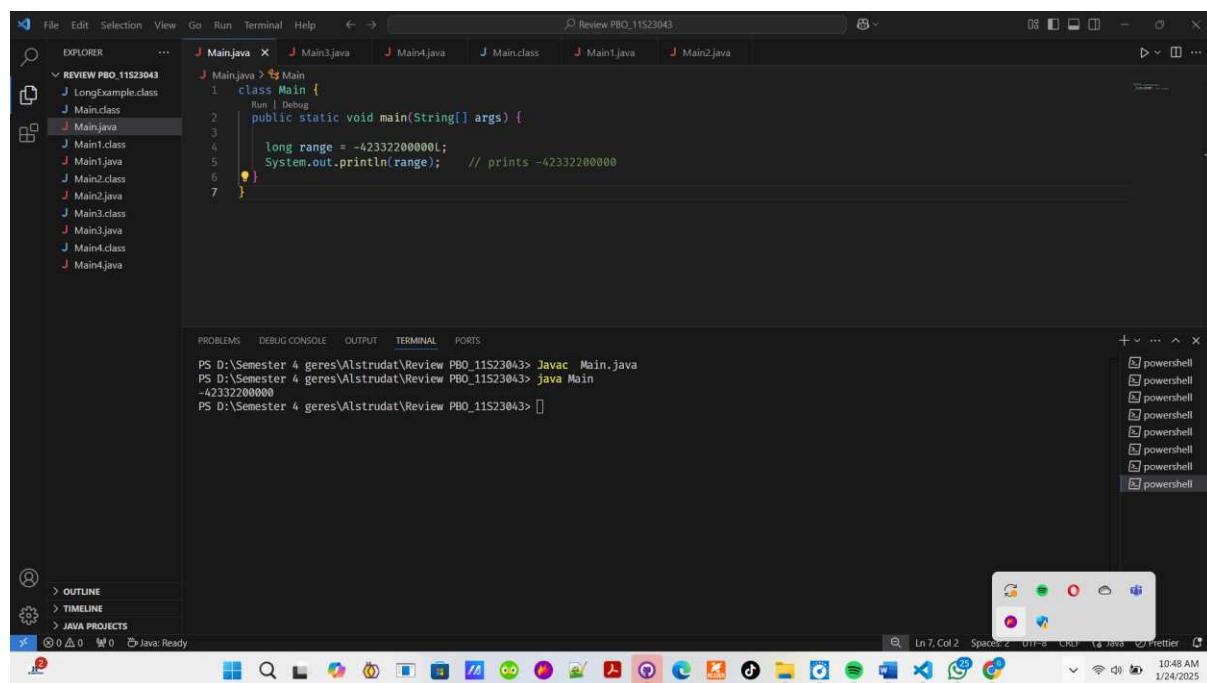
The terminal output shows the execution of the Java command and the resulting output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main4
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main4
-4250000
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java long data type

11S2215 - Algorithms and Data Structures

Laporan Praktikum



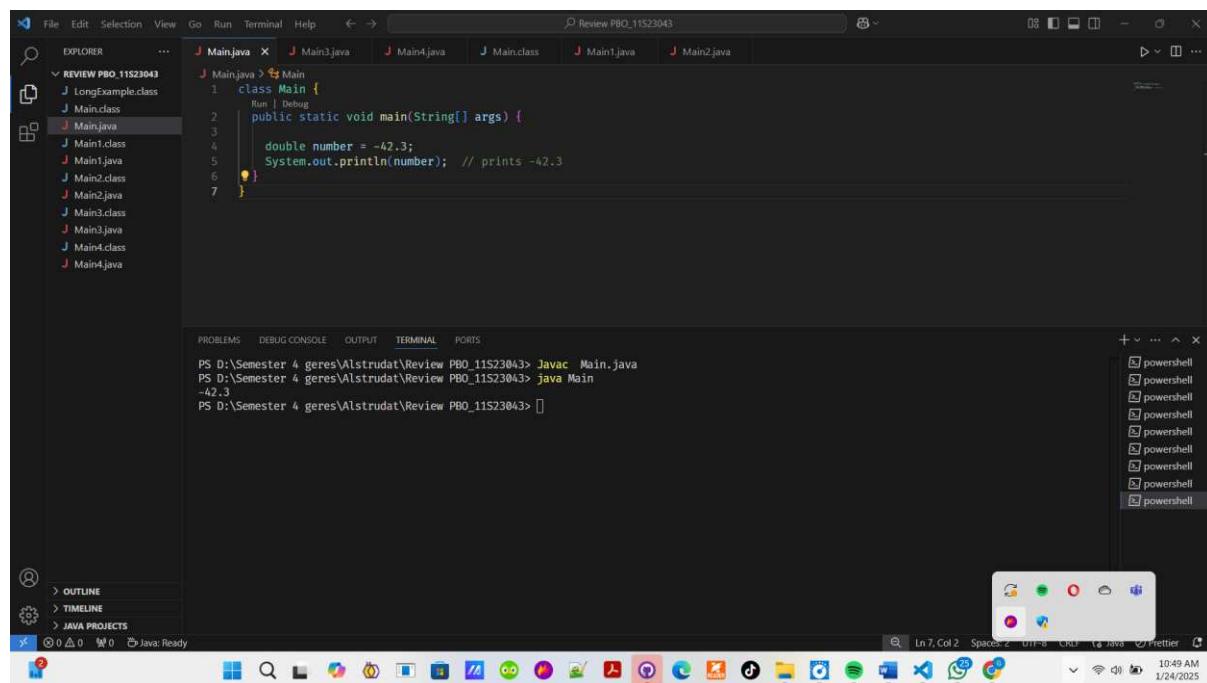
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing files: LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, and Main4.java.
- Code Editor:** Displays the "Main.java" file with the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         long range = -42332200000L;  
4         System.out.println(range); // prints -42332200000  
5     }  
6 }
```
- Terminal:** Shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
-42332200000  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows "Java Ready".

Java double data type



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing files: LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, and Main4.java.
- Code Editor:** Displays the "Main.java" file with the following code:

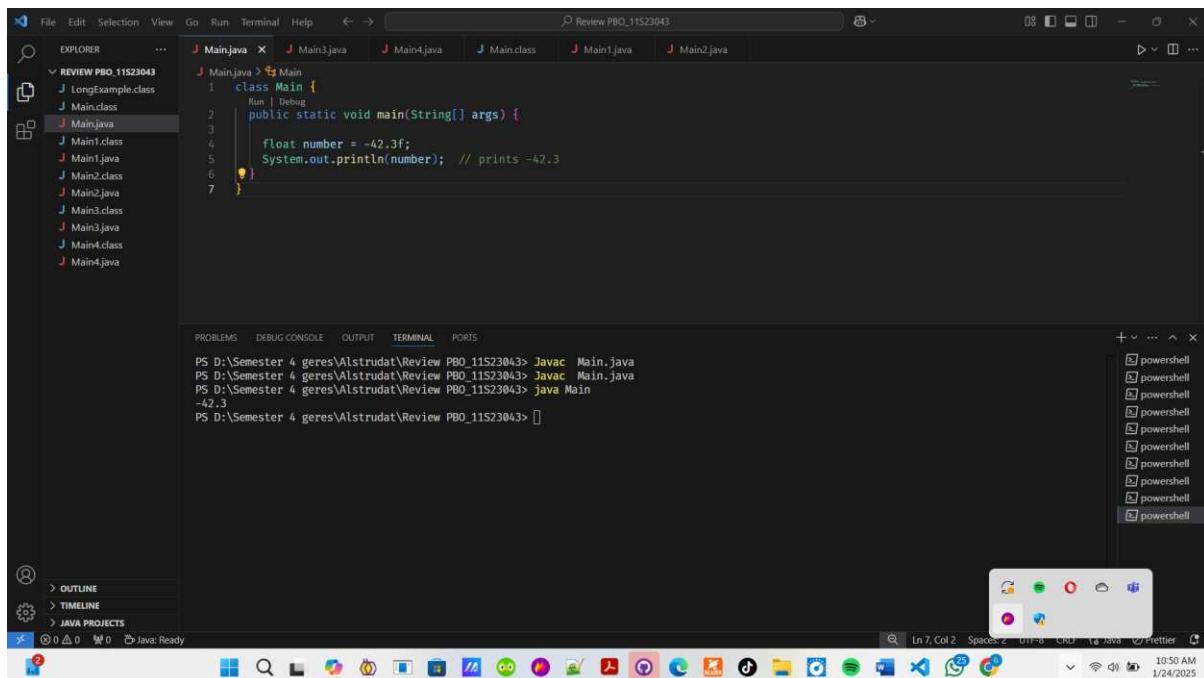
```
1 class Main {  
2     public static void main(String[] args) {  
3         double number = -42.3;  
4         System.out.println(number); // prints -42.3  
5     }  
6 }
```
- Terminal:** Shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
-42.3  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows "Java Ready".

Java float data type

11S2215 - Algorithms and Data Structures

Laporan Praktikum



The screenshot shows a Java development environment, likely IntelliJ IDEA or a similar IDE, running on a Windows operating system. The code editor displays a file named Main.java with the following content:

```
class Main {
    public static void main(String[] args) {
        float number = -42.3f;
        System.out.println(number); // prints -42.3
    }
}
```

The terminal window below shows the command-line output of the Java application:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
-42.3
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

The taskbar at the bottom of the screen shows various open applications, including Microsoft Edge, File Explorer, and other productivity tools.

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Java char data type

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file is open, containing the following code:

```
class Main {
    public static void main(String[] args) {
        char letter = '\u0051';
        System.out.println(letter); // prints Q
    }
}
```

The terminal window shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Q
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

✓ Java Operators

Arithmetic Operators

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file is open, containing the following code:

```
class Main {
    public static void main(String[] args) {
        // declare variables
        int a = 12, b = 5;

        // addition operator
        System.out.println("a + b = " + (a + b));

        // subtraction operator
        System.out.println("a - b = " + (a - b));

        // multiplication operator
        System.out.println("a * b = " + (a * b));

        // division operator
        System.out.println("a / b = " + (a / b));

        // modulo operator
        System.out.println("a % b = " + (a % b));
    }
}
```

The terminal window shows the output of running the code:

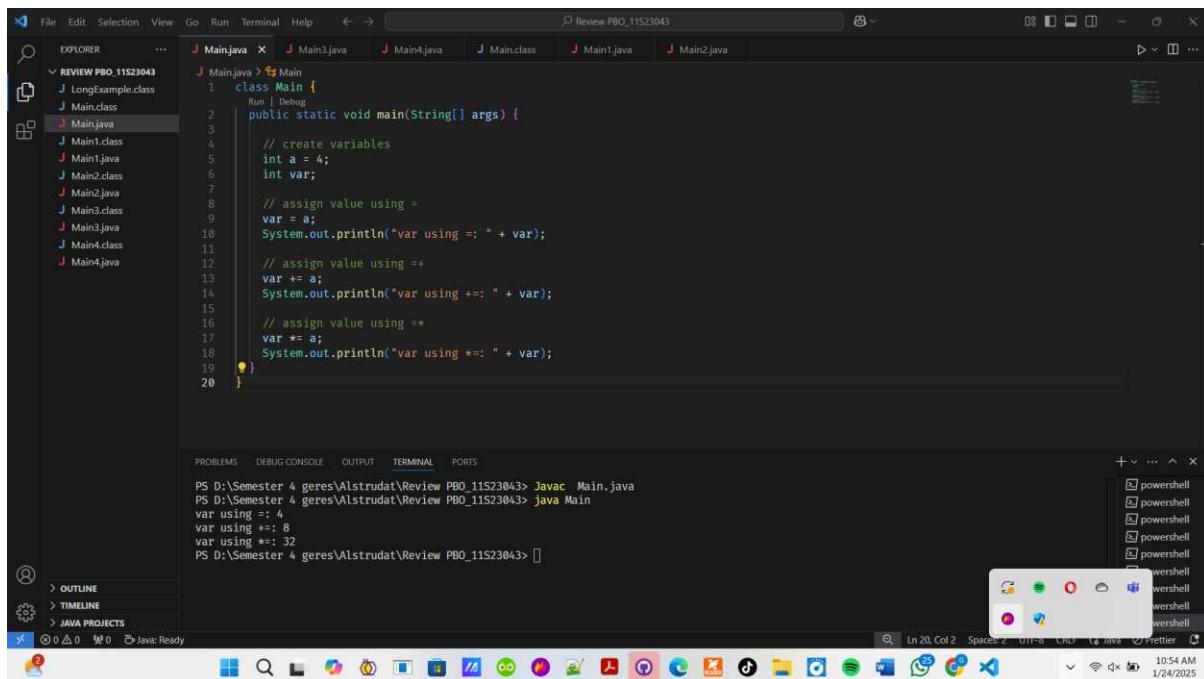
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
a + b = 17
a - b = 7
a * b = 60
a / b = 2
a % b = 2
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Assignment Operators



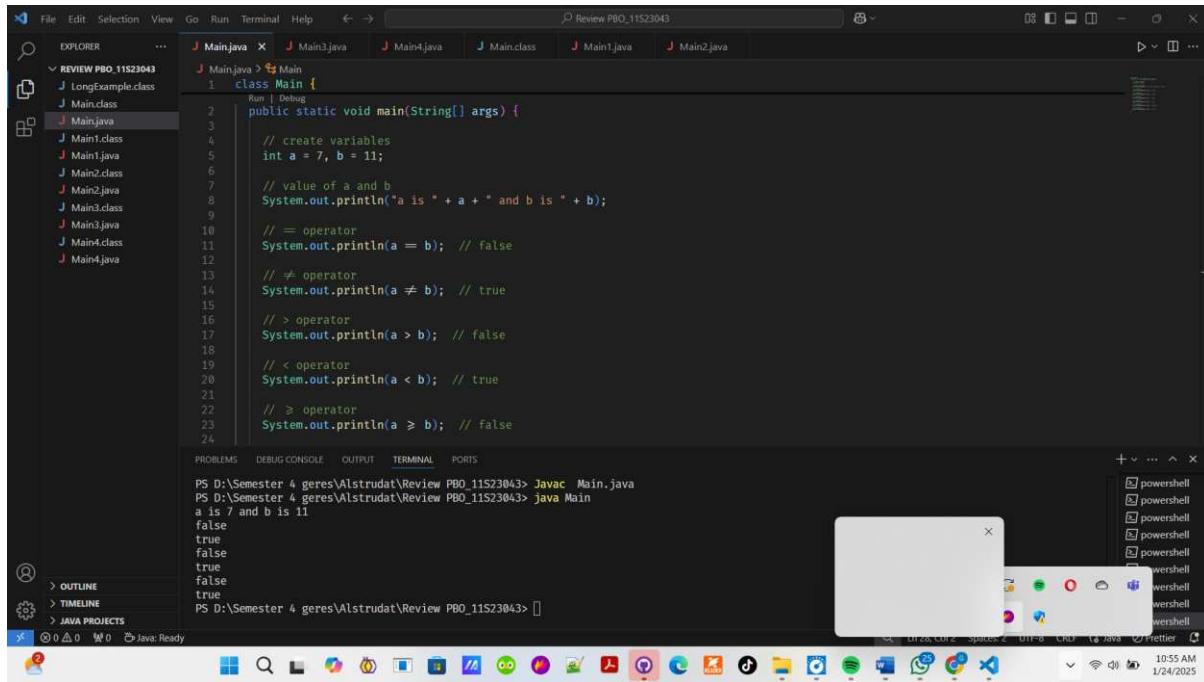
A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project named "REVIEW PBO_11S23043" containing several Java files: LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, and Main4.java. The "Main.java" file is open in the editor, displaying the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         // create variables  
4         int a = 4;  
5         int var;  
6  
7         // assign value using =  
8         var = a;  
9         System.out.println("var using := " + var);  
10  
11        // assign value using +=  
12        var += a;  
13        System.out.println("var using +=: " + var);  
14  
15        // assign value using *=  
16        var *= a;  
17        System.out.println("var using *=: " + var);  
18  
19    }  
20 }
```

The terminal at the bottom shows the command-line output of running the Java code:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main  
var using := 4  
var using += 8  
var using *= 32
```

Relational Operators



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project named "REVIEW PBO_11S23043" containing several Java files: LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, and Main4.java. The "Main.java" file is open in the editor, displaying the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         // create variables  
4         int a = 7, b = 11;  
5  
6         // value of a and b  
7         System.out.println("a is " + a + " and b is " + b);  
8  
9         // = operator  
10        System.out.println(a == b); // false  
11  
12        // != operator  
13        System.out.println(a != b); // true  
14  
15        // > operator  
16        System.out.println(a > b); // false  
17  
18        // < operator  
19        System.out.println(a < b); // true  
20  
21        // ≥ operator  
22        System.out.println(a ≥ b); // false  
23  
24    }  
25 }
```

The terminal at the bottom shows the command-line output of running the Java code:

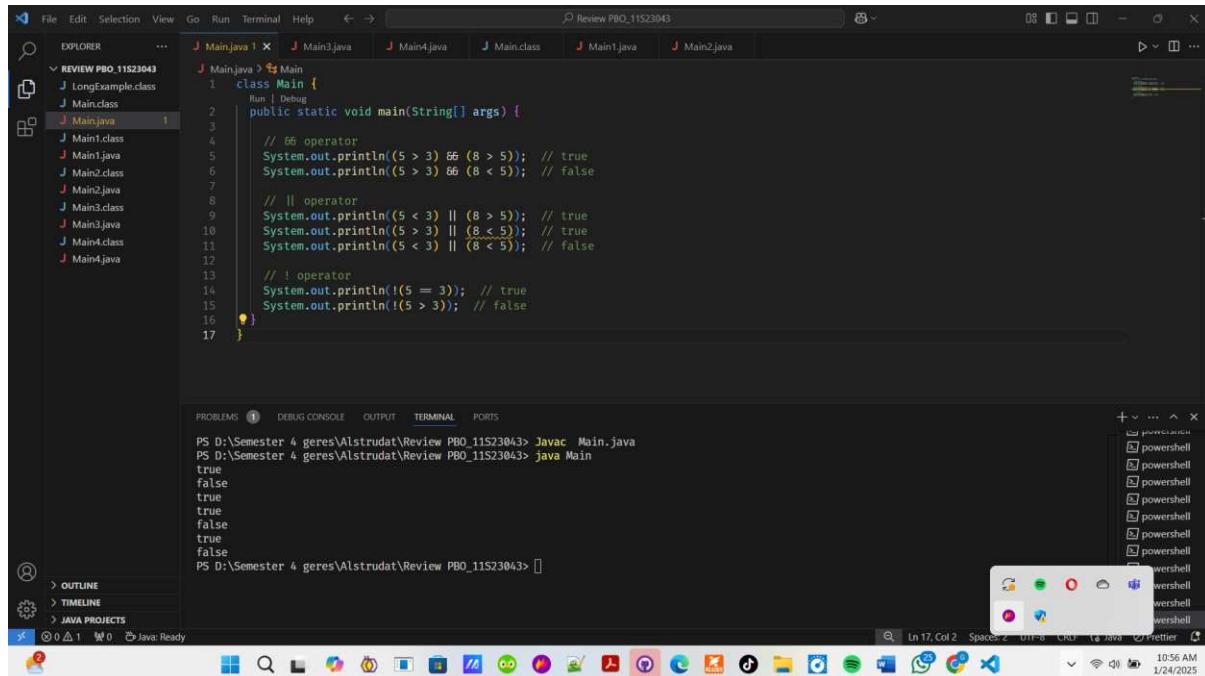
```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main  
a is 7 and b is 11  
false  
true  
false  
true  
false  
true
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Logical Operators



A screenshot of the Visual Studio Code (VS Code) interface. The code editor shows a Java file named Main.java with the following code:

```
class Main {
    public static void main(String[] args) {
        // && operator
        System.out.println((5 > 3) && (8 > 5)); // true
        System.out.println((5 > 3) && (8 < 5)); // false

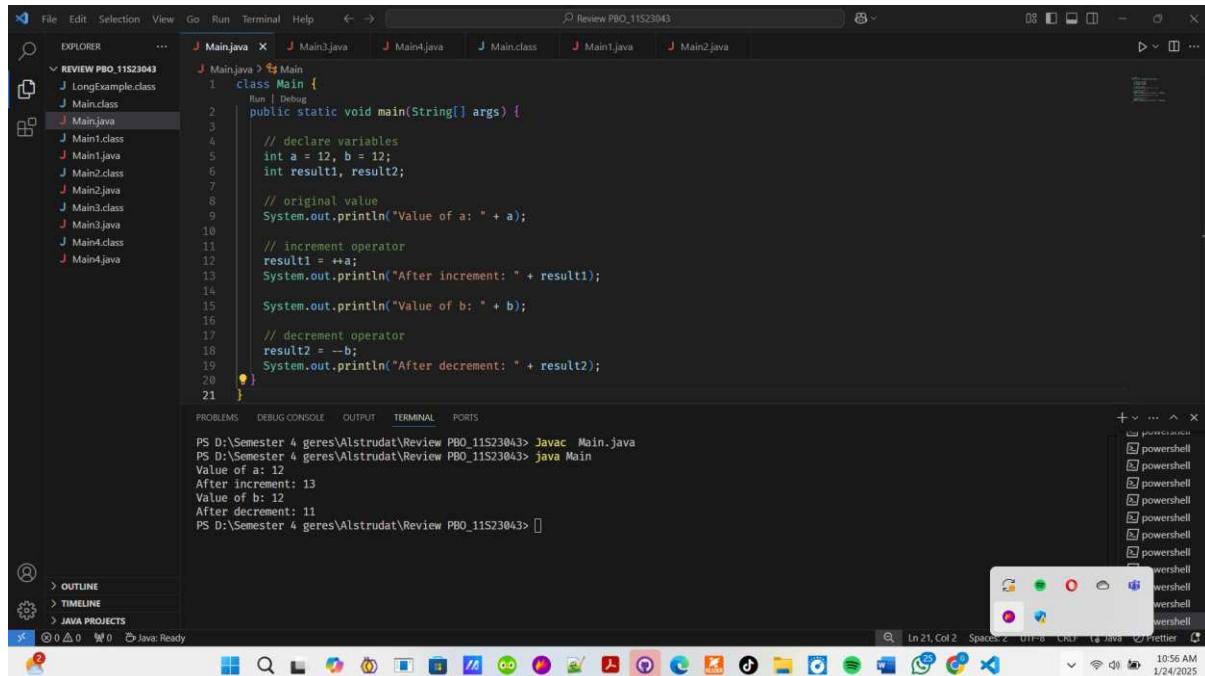
        // || operator
        System.out.println((5 < 3) || (8 > 5)); // true
        System.out.println((5 > 3) || (8 < 5)); // true
        System.out.println((5 < 3) || (8 < 5)); // false

        // ! operator
        System.out.println(!(5 == 3)); // true
        System.out.println(!(5 > 3)); // false
    }
}
```

The terminal below shows the output of the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
true
false
true
true
false
true
false
true
false
true
false
true
false
true
false
```

Increment and Decrement Operators



A screenshot of the Visual Studio Code (VS Code) interface. The code editor shows a Java file named Main.java with the following code:

```
class Main {
    public static void main(String[] args) {
        // declare variables
        int a = 12, b = 12;
        int result1, result2;

        // original value
        System.out.println("Value of a: " + a);

        // increment operator
        result1 = ++a;
        System.out.println("After increment: " + result1);

        System.out.println("Value of b: " + b);

        // decrement operator
        result2 = --b;
        System.out.println("After decrement: " + result2);
    }
}
```

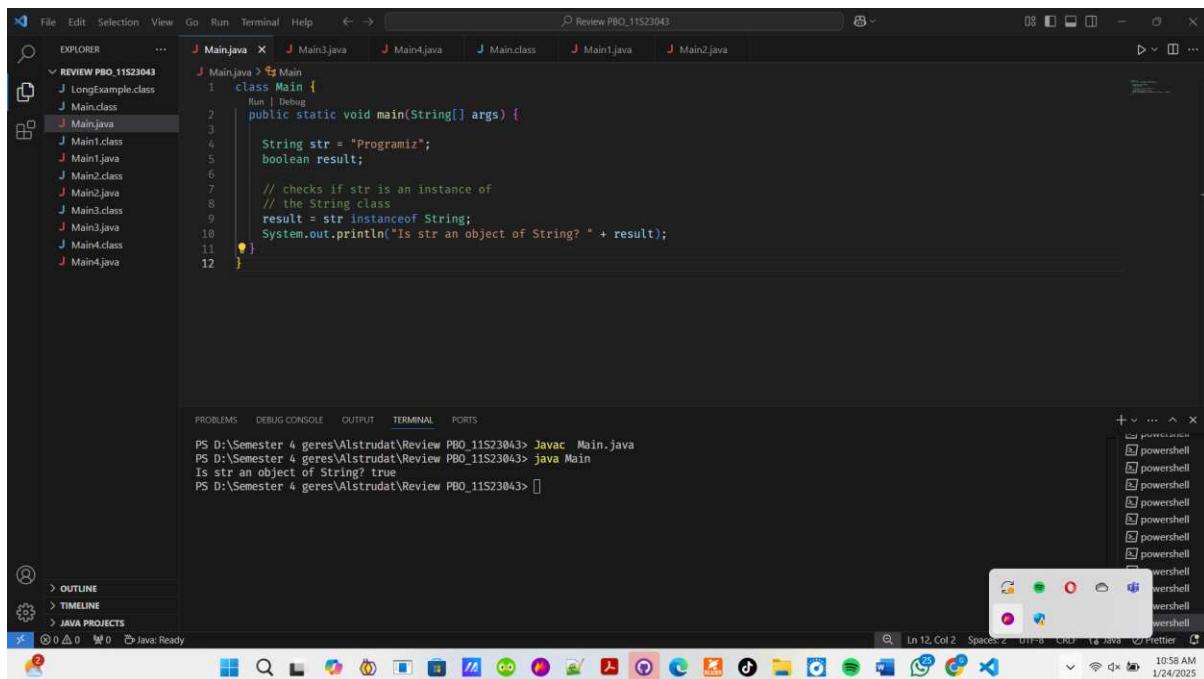
The terminal below shows the output of the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Value of a: 12
After increment: 13
Value of b: 12
After decrement: 11
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

Java instanceof Operator



```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Review PBO_11S23043 J Main.java X J Main3.java J Main4.java J Main.class J Main1.java J Main2.java J LongExample.class J Main.java
1 class Main {
2     Run | Debug
3
4     public static void main(String[] args) {
5
6         String str = "Programiz";
7         boolean result;
8
9         // checks if str is an instance of
10        // the String class
11        result = str instanceof String;
12        System.out.println("Is str an object of String? " + result);
13    }
14}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

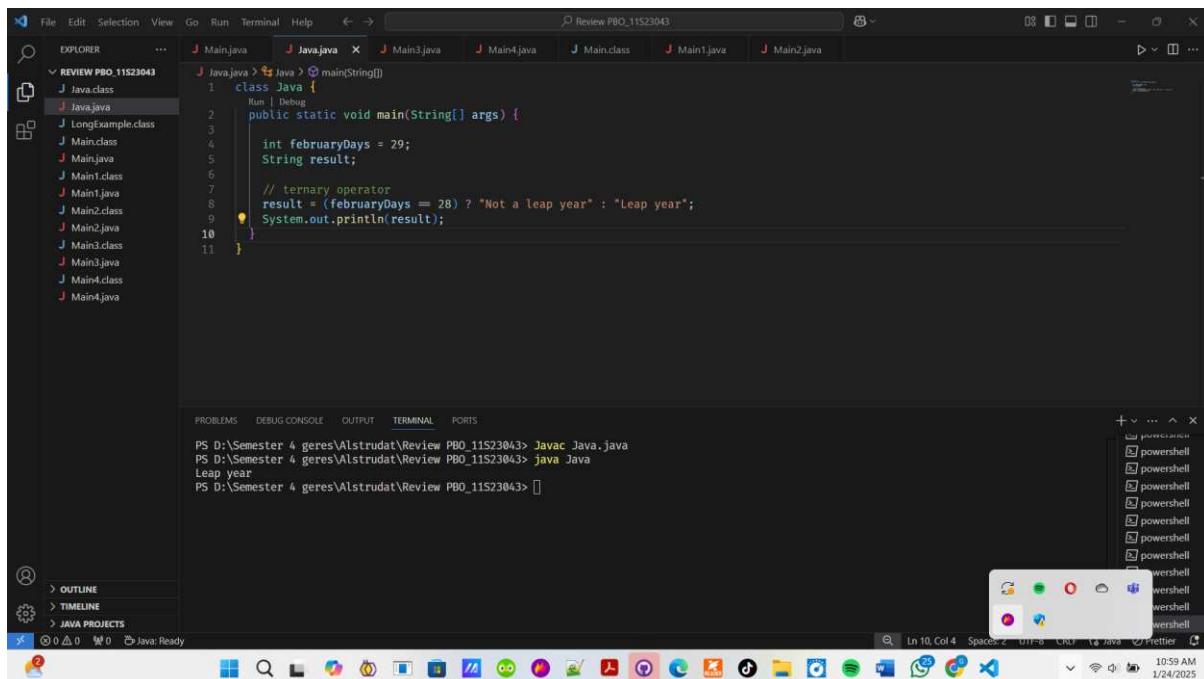
```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Is str an object of String? true
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

OUTLINE TIMELINE JAVA PROJECTS

Java Ready

Ln 12, Col 2 Spaces: 2 011-5 CPU 10:58 AM 1/24/2025

Java Ternary Operator



```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Review PBO_11S23043 J Main.java J Java.java X J Main3.java J Main4.java J Main.class J Main1.java J Main2.java J LongExample.class J Java.java
1 class Java {
2     Run | Debug
3
4     public static void main(String[] args) {
5
6         int februaryDays = 29;
7         String result;
8
9         // ternary operator
10        result = (februaryDays == 28) ? "Not a leap year" : "Leap year";
11        System.out.println(result);
12    }
13}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Java.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Java
Leap year
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

OUTLINE TIMELINE JAVA PROJECTS

Java Ready

Ln 10, Col 4 Spaces: 2 011-5 CPU 10:59 AM 1/24/2025

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Challenge

A screenshot of the Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "Review PBO_11S23043". The Explorer sidebar shows a project named "REVIEW PBO_11S23043" containing several Java files: Main.java, Challenge.java, Java.java, Main3.java, Main4.java, Main.class, Main1.java, Main1.class, Main2.java, Main2.class, Main3.java, Main3.class, Main4.java, and Main4.class. The main editor tab displays the "Challenge.java" file with the following code:

```
public class Challenge {
    public static int findLargest(int num1, int num2) {
        return (num1 > num2) ? num1 : num2;
    }

    public static void main(String[] args) {
        int num1 = 4;
        int num2 = 5;
        System.out.println("The largest number is: " + findLargest(num1, num2));
    }
}
```

The terminal below the editor shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Challenge.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Challenge
The largest number is: 5
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

The status bar at the bottom indicates "Java: Ready".

✓ Java Basic Input and Output

Java Output

A screenshot of the Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "Review PBO_11S23043". The Explorer sidebar shows a project named "REVIEW PBO_11S23043" containing several Java files: Main.java, AssignmentOperator.java, Java.java, Main3.java, Main4.java, Main.class, Main1.java, Main1.class, Main2.java, Main2.class, Main3.java, Main3.class, Main4.java, and Main4.class. The main editor tab displays the "AssignmentOperator.java" file with the following code:

```
class AssignmentOperator {
    public static void main(String[] args) {
        System.out.println("Java programming is interesting.");
    }
}
```

The terminal below the editor shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac AssignmentOperator.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java AssignmentOperator
Java programming is interesting.
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

The status bar at the bottom indicates "Java: Ready".

11S2215 - Algorithms and Data Structures

Laporan Praktikum



print() and println()

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing files like Main.java, Challenge.java, AssignmentOperator.java, Output.java, Java.java, Main3.java, Main4.java, Main.class, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, and Output.class.
- Code Editor:** Displays the "Output.java" file with the following code:

```
1 class Output {
2     public static void main(String[] args) {
3         System.out.println("1. println ");
4         System.out.println("2. println ");
5         System.out.print("1. print ");
6         System.out.print("2. print");
7     }
8 }
```
- Terminal:** Shows the command-line output of running the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Output.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Output
1. println
2. println
1. print 2. print
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> 
```
- Bottom Status Bar:** Shows the date and time as 11:09 AM, 1/24/2025.

Printing Variables and Literals

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing files like Main.java, Challenge.java, AssignmentOperator.java, Variables.java, Java.java, Main3.java, Main4.java, Main.class, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Output.class, and Output.java.
- Code Editor:** Displays the "Variables.java" file with the following code:

```
1 class Variables {
2     public static void main(String[] args) {
3         Double number = -10.6;
4         System.out.println(5);
5         System.out.println(number);
6     }
7 }
```
- Terminal:** Shows the command-line output of running the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Variables.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Variables
5
-10.6
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> 
```
- Bottom Status Bar:** Shows the date and time as 11:11 AM, 1/24/2025.

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Print Concatenated Strings

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing several Java files: Main.java, Challenge.java, AssignmentOperator.java, Output.java, Variables.java, PrintVariables.java, Java.java, Main3.java, Main4.java, LongExample.class, Main.class, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Output.class, Output.java, and PrintVariables.class.
- Code Editor:** Displays the "PrintVariables.java" file with the following code:

```
1 class PrintVariables {
2     public static void main(String[] args) {
3         Double number = -10.6;
4         System.out.println("I am " + "awesome.");
5         System.out.println("Number = " + number);
6     }
7 }
```
- Terminal:** Shows the output of running the Java compiler (javac) and the Java runtime (java) on the "PrintVariables.java" file. The terminal window shows:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac PrintVariables.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java PrintVariables
I am awesome.
Number = -10.6
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows the current time as 11:13 AM and the date as 1/24/2025.

Java Input

Get Integer Input From the User

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing several Java files: Main.java, Challenge.java, AssignmentOperator.java, Output.java, Variables.java, Input.java, Java.java, Main3.java, Main4.java, LongExample.class, Main.class, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Output.class, Output.java, PrintVariables.class, PrintVariables.java, Variables.class, and Variables.java.
- Code Editor:** Displays the "Input.java" file with the following code:

```
1 import java.util.Scanner;
2
3 class Input {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         System.out.print("Enter an integer: ");
7         int number = input.nextInt();
8         System.out.println("You entered " + number);
9         // closing the scanner object
10        input.close();
11    }
12 }
```
- Terminal:** Shows the output of running the Java compiler (javac) and the Java runtime (java) on the "Input.java" file. The terminal window shows:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Input.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Input
Enter an integer: 19
You entered 19
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows the current time as 11:13 AM and the date as 1/24/2025.

11S2215 - Algorithms and Data Structures

Laporan Praktikum

Get float, double and String Input

✓ Java Expressions, Statements and Blocks

Java Blocks

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

- File Explorer:** Shows a project named "REVIEW PBO_11S2043" containing numerous Java files, with "Main.java" currently selected.
- Code Editor:** Displays the "Main.java" code:

```
1 class Main {
2     public static void main(String[] args) {
3
4         String band = "Beatles";
5
6         if (band == "Beatles") { // start of block
7             System.out.print("Hey ");
8             System.out.print("Grace!");
9         } // end of block
10    }
11 }
```
- Terminal:** Shows the command-line interface output:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S2043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S2043> java Main
Hey Grace!
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S2043> 
```
- Bottom Status Bar:** Shows "Java: Ready".
- System Tray:** Shows icons for power shell, task manager, file explorer, and other system utilities.

The screenshot shows a Java code editor interface with the following details:

- File Explorer:** On the left, it lists several Java files under the project "REVIEW PBO_11S23043". The file "Main.java" is currently selected.
- Code Editor:** The main area displays the "Main.java" code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         if (10 > 5) { // start of block  
4             // body of block  
5         } // end of block  
6     }  
7 }  
8 }
```
- Terminal:** At the bottom, a terminal window shows the command-line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]
```
- Bottom Bar:** The taskbar at the bottom shows various pinned icons for Microsoft Office applications like Word, Excel, and Powerpoint, as well as browser icons for Edge, Chrome, and Firefox.

- Java Flow Control

✓ Java if (if-then) Statement

Java if Statement

The screenshot shows a Java project structure on the left with files like Main.java, IfStatement.java, AssignmentOperator.java, etc. The IfStatement.java file is open in the editor, containing a simple if-statement. The terminal at the bottom shows the command `java IfStatement` being run, which outputs "Statement outside if block".

```
File Edit Selection View Go Run Terminal Help < > Review PBO_11S23043
EXPLORER
REVIEW PBO_11S23043
AssignmentOperator...
AssignmentOperator...
Challenge.class
Challenge.java
IfStatement.class
IfStatement.java
Input.class
Input.java
Java.class
Java.java
LongExample.class
Main.class
Main.java
Main1.class
Main1.java
Main2.class
Main2.java
Main3.class
Main3.java
Main4.class
Main4.java
Output.class
Output.java
PrintVariables.class
PrintVariables.java
Variables.class
Variables.java

J Main.java J IfStatement.java x J Challenge.java J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja ...
J #Statement.java > IfStatement > main(String[])
1 class IfStatement {
2     Run | Debug
3         public static void main(String[] args) {
4
5             int number = 10;
6
7             // checks if number is less than 0
8             if (number < 0) {
9                 System.out.println("The number is negative.");
10            }
11            System.out.println("Statement outside if block");
12        }
13    }

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac IfStatement.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java IfStatement
Statement outside if block
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [ ]
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Java if with String

The screenshot shows a Java code editor in VS Code. The current file is `Main.java`, which contains the following code:

```
1 class Main {
2     public static void main(String[] args) {
3         // create a string variable
4         String language = "Java";
5
6         // if statement
7         if (language == "Java") {
8             System.out.println("Best Programming Language");
9         }
10    }
11 }
```

The terminal below shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Best Programming Language
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java if...else (if-then-else) Statement

Java if Statement

The screenshot shows a Java code editor in VS Code. The current file is `IfStatement.java`, which contains the following code:

```
1 class IfStatement {
2     public static void main(String[] args) {
3
4         int number = 10;
5
6         // checks if number is less than 0
7         if (number < 0) {
8             System.out.println("The number is negative.");
9         }
10        System.out.println("Statement outside if block");
11    }
12 }
```

The terminal below shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac IfStatement.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java IfStatement
Statement outside if block
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Java if with String

The screenshot shows a Java file named Main.java in the Explorer panel. The code contains an if statement that prints "Best Programming Language" if the variable language is set to "Java". The terminal window shows the output of running the Java compiler (javac) and the Java interpreter (java) on the Main.java file, which prints "Best Programming Language".

```
1 class Main {
2     public static void main(String[] args) {
3         // create a string variable
4         String language = "Java";
5
6         // if statement
7         if (language == "Java") {
8             System.out.println("Best Programming Language");
9         }
10    }
11 }
```

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Best Programming Language
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java if...else (if-then-else) Statement

Java if...else Statement

The screenshot shows a Java file named Main.java in the Explorer panel. The code uses an if...else statement to check if a number is greater than 0. If it is, it prints "The number is positive.". Otherwise, it prints "The number is not positive.". The terminal window shows the output of running the Java compiler (javac) and the Java interpreter (java) on the Main.java file, which prints "The number is positive.".

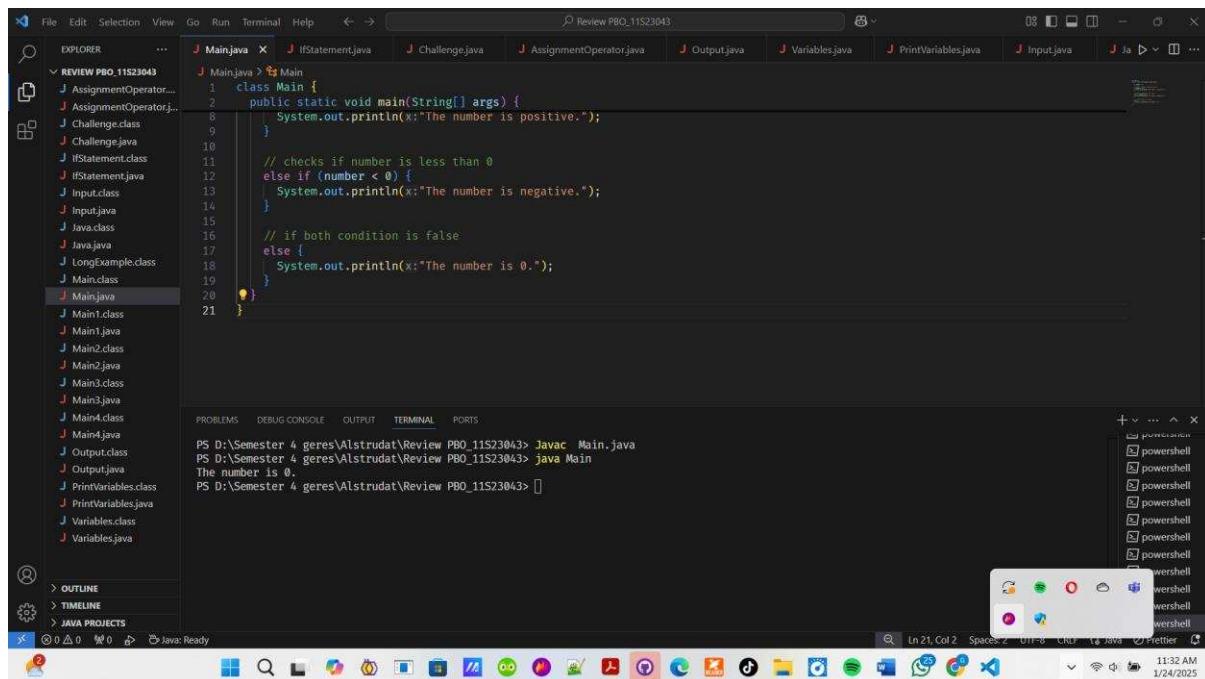
```
1 class Main {
2     public static void main(String[] args) {
3         int number = 10;
4
5         // checks if number is greater than 0
6         if (number > 0) {
7             System.out.println("The number is positive.");
8         }
9
10        // execute this block
11        // if number is not greater than 0
12        else {
13            System.out.println("The number is not positive.");
14        }
15
16        System.out.println("Statement outside if...else block");
17    }
18 }
```

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The number is positive.
Statement outside if...else block
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

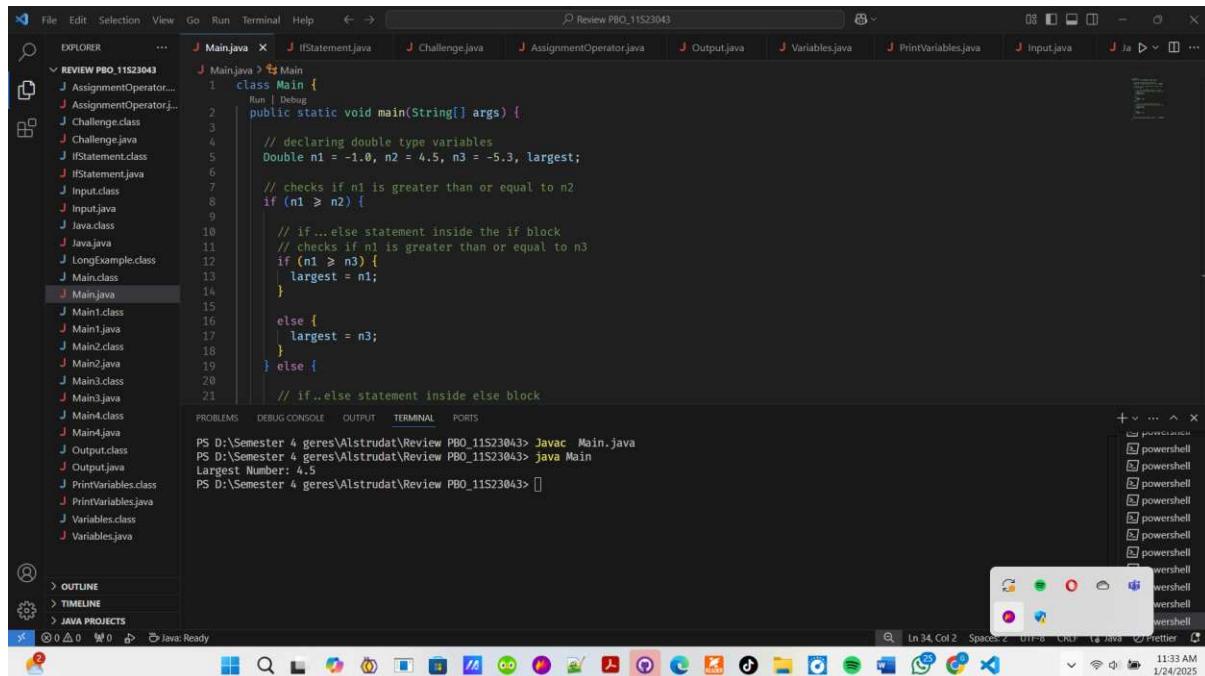
Java if...else...if Statement



```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER J Main.java X J IfStatement.java J Challenge.java J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja ...
J AssignmentOperator...
J AssignmentOperator...
J Challenge.class
J Challenge.java
J IfStatement.class
J IfStatement.java
J Input.class
J Input.java
J Java.class
J Java.java
J LongExample.class
J Main.class
J Main.java
J Main1.class
J Main1.java
J Main2.class
J Main2.java
J Main3.class
J Main3.java
J Main4.class
J Main4.java
J Output.class
J Output.java
J PrintVariables.class
J PrintVariables.java
J Variables.class
J Variables.java
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The number is 0.
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
OUTLINE TIMELINE JAVA PROJECTS
Java: Ready
11:32 AM 1/24/2025
```

Java Nested if..else Statement

Nested if...else Statement



```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER J Main.java X J IfStatement.java J Challenge.java J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja ...
J AssignmentOperator...
J AssignmentOperator...
J Challenge.class
J Challenge.java
J IfStatement.class
J IfStatement.java
J Input.class
J Input.java
J Java.class
J Java.java
J LongExample.class
J Main.class
J Main.java
J Main1.class
J Main1.java
J Main2.class
J Main2.java
J Main3.class
J Main3.java
J Main4.class
J Main4.java
J Output.class
J Output.java
J PrintVariables.class
J PrintVariables.java
J Variables.class
J Variables.java
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Largest Number: 4.5
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
OUTLINE TIMELINE JAVA PROJECTS
Java: Ready
11:33 AM 1/24/2025
```

Challenge

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it lists several Java files under the project "REVIEW PBO_11S23043".
- Code Editor:** The main area displays the content of the `PassOrFail.java` file.
- Terminal:** At the bottom, a terminal window shows the command-line output of running the `PassOrFail` class.

```
File Edit Selection View Go Run Terminal Help < > Review PBO_11S23043
EXPLORER J Main.java J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja ...
REVIEW PBO_11S23043
J AssignmentOperator...
J AssignmentOperator...
J Challenge.class
J IfStatement.class
J IfStatement.java
J Input.class
J Input.java
J Java.class
J Java.java
J LongExample.class
J Main.class
J Main.java
J Main1.class
J Main1.java
J Main2.class
J Main2.java
J Main3.class
J Main3.java
J Main4.class
J Main4.java
J Output.class
J Output.java
J PassOrFail.class
J PassOrFail.java
J PrintVariables.class
J PrintVariables.java
J Variables.class
J Variables.java

Main.java
J PassOrFail.java > PassOrFail
1 public class PassOrFail {
2     public static String checkPassOrFail(int score) {
3         return (score >= 50) ? "Pass" : "Fail";
4     }
5
6     Run | Debug
7     public static void main(String[] args) {
8         int score = 55; // Example score
9         System.out.println("Result: " + checkPassOrFail(score)); // Output: Pass
10    }
11

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac PassOrFail.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java PassOrFail
Result: Pass
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [ ]
```

The terminal output shows the successful compilation of `PassOrFail.java` and its execution, which prints "Result: Pass".

✓ Java Ternary Operator

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

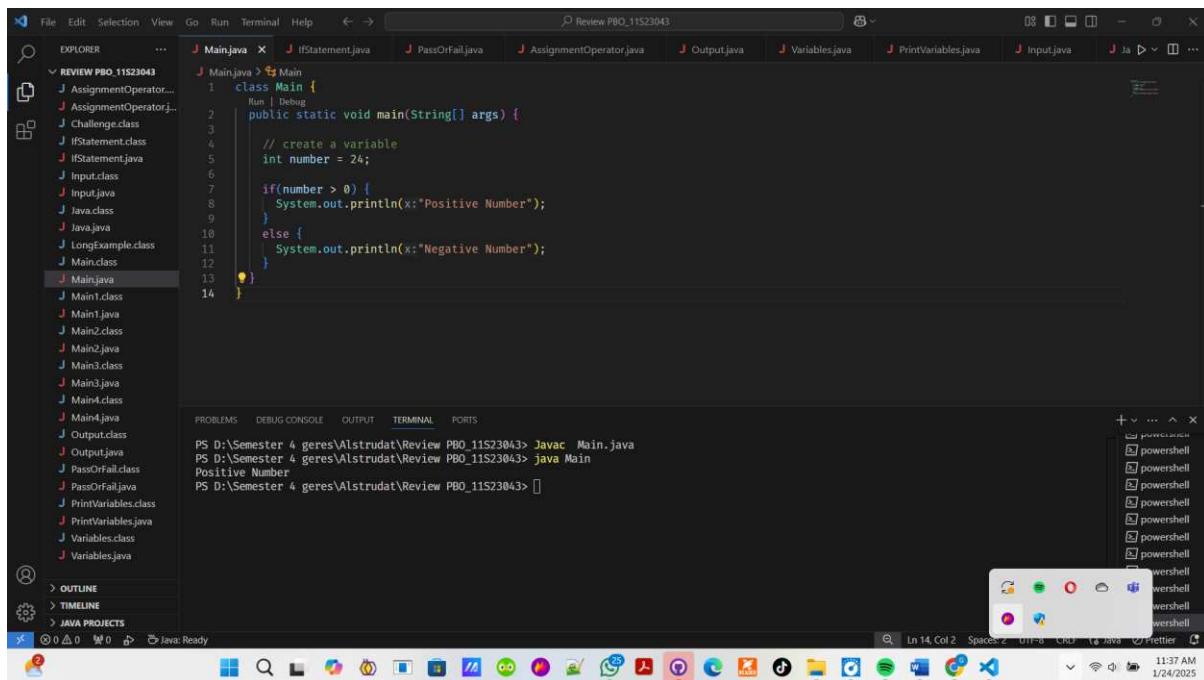
- File Explorer:** Shows a project named "REVIEW PBO_11S2043" containing various Java files: AssignmentOperator..., AssignmentOperator..., Challenge.class, iStatement.class, iStatement.java, Input.class, Input.java, Java.class, Java.java, LongExample.class, Main.class, Main.java (selected), Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Output.class, Output.java, PassOrFail.class, PassOrFail.java, PrintVariables.class, PrintVariables.java, Variables.class, Variables.java.
- Code Editor:** The main editor window displays the "Main.java" source code:

```
1 import java.util.Scanner;
2
3 class Main {
4     Run | Debug
5     public static void main(String[] args) {
6
7         // take input from users
8         Scanner input = new Scanner(System.in);
9         System.out.println("Enter your marks: ");
10        double marks = input.nextDouble();
11
12        // ternary operator checks if
13        // marks is greater than 40
14        String result = (marks > 40) ? "pass" : "fail";
15
16        System.out.println("You " + result + " the exam.");
17        input.close();
18    }
19 }
```
- Terminal:** The terminal window shows the execution of the Java program:

```
Enter your marks:
99
You pass the exam.
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Enter your marks:
44
You pass the exam.
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Enter your marks:
30
You fail the exam.
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Bar:** Includes icons for file operations (New, Open, Save, Find, Copy, Paste, Delete), Java Ready status, and system status indicators (CPU, RAM, Disk, Network).

11S2215 - Algorithms and Data Structures

Laporan Praktikum

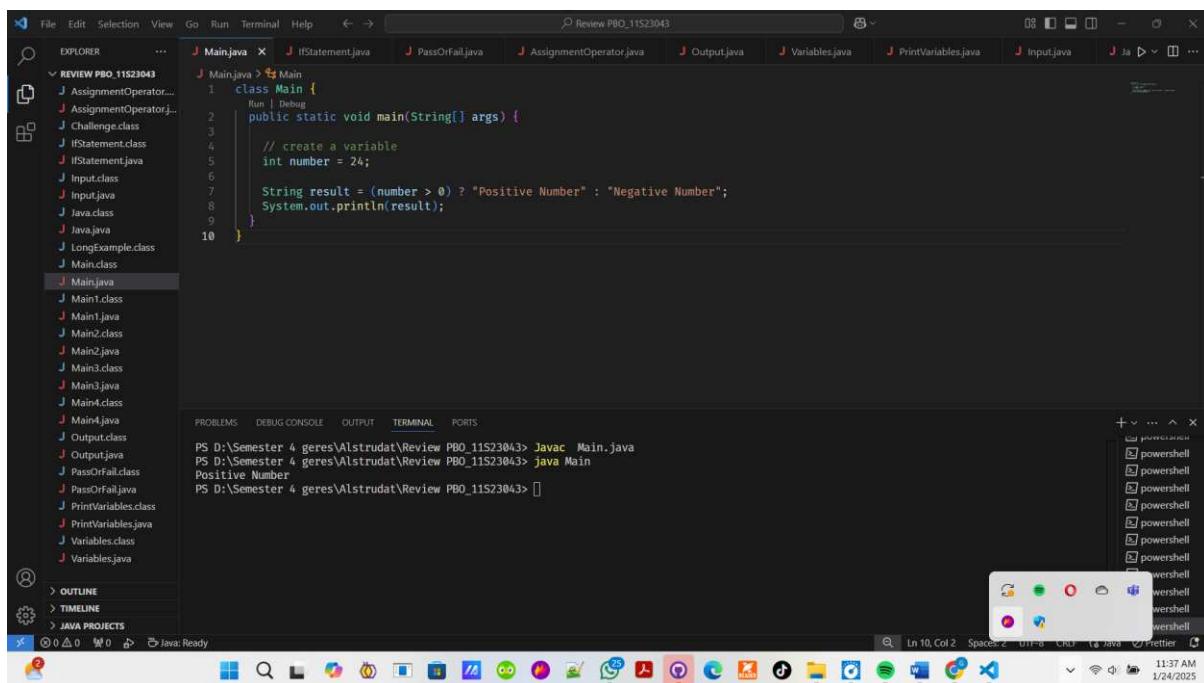


The screenshot shows a Java file named Main.java in the VS Code editor. The code contains a main method that prints "Positive Number" if a variable 'number' is greater than 0, and "Negative Number" otherwise. The terminal below shows the output of the Java compiler (javac) and the Java runtime (java) both printing "Positive Number".

```
1 class Main {  
2     public static void main(String[] args) {  
3         // create a variable  
4         int number = 24;  
5  
6         if(number > 0) {  
7             System.out.println("Positive Number");  
8         }  
9         else {  
10            System.out.println("Negative Number");  
11        }  
12    }  
13}  
14
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
Positive Number  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> 
```



The screenshot shows the same Main.java file with a different implementation. Instead of using an if-else block, it uses a ternary operator to determine the string result based on the value of 'number'. The terminal output remains the same.

```
1 class Main {  
2     public static void main(String[] args) {  
3         // create a variable  
4         int number = 24;  
5  
6         String result = (number > 0) ? "Positive Number" : "Negative Number";  
7         System.out.println(result);  
8     }  
9 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
Positive Number  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> 
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Nested Ternary Operators

The screenshot shows the VS Code interface with the file `Main.java` open. The code contains a nested ternary operator:public static void main(String[] args) {
 int n1 = 2, n2 = 9, n3 = -11;

 int largest = (n1 >= n2) ? (n1 >= n3) ? n1 : n3 : ((n2 >= n3) ? n2 : n3);
 System.out.println("Largest Number: " + largest);
}The terminal tab shows the output of running the Java code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
Largest Number: 9  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Challenge

The screenshot shows the VS Code interface with the file `Main.java` open. The code contains an if-statement inside the main method:

```
public class Main {  
    public static String checkEntry(int age) {  
        return (age >= 18) ? "Can Enter" : "Cannot Enter";  
    }  
  
    public static void main(String[] args) {  
        int age = 21; // Example age  
        System.out.println("Result: " + checkEntry(age)); // Output: Can Enter  
    }  
}
```

The terminal tab shows the output of running the Java code:

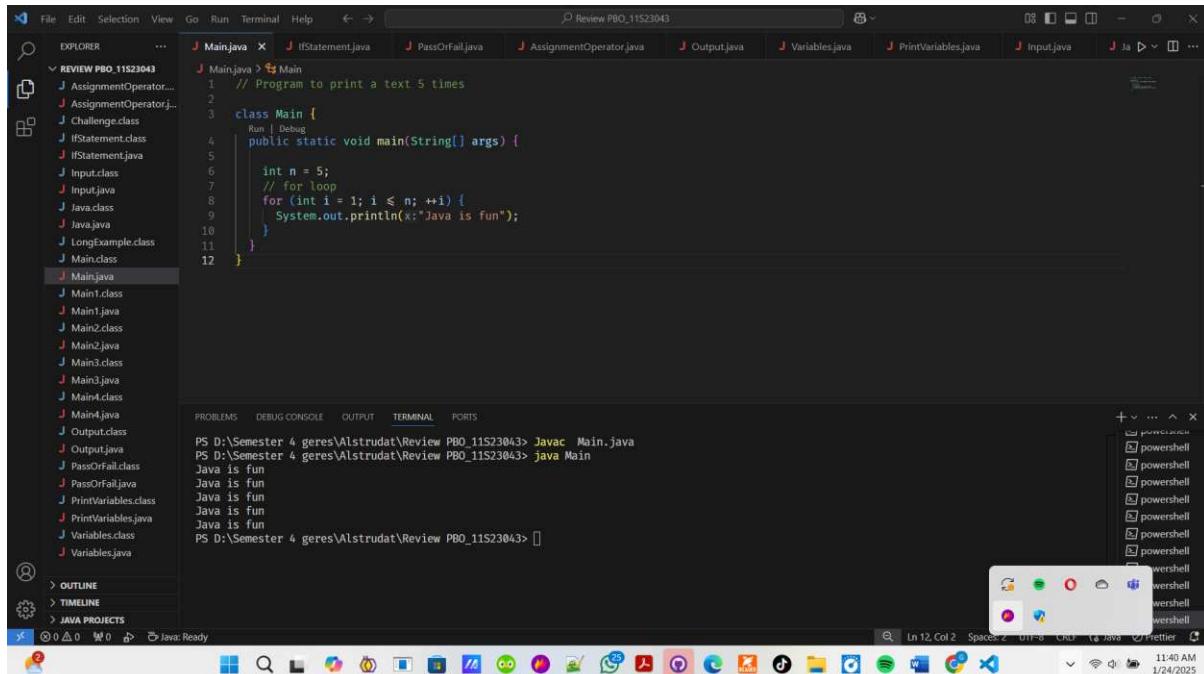
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
Result: Can Enter  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java for Loop

Display a Text Five Times

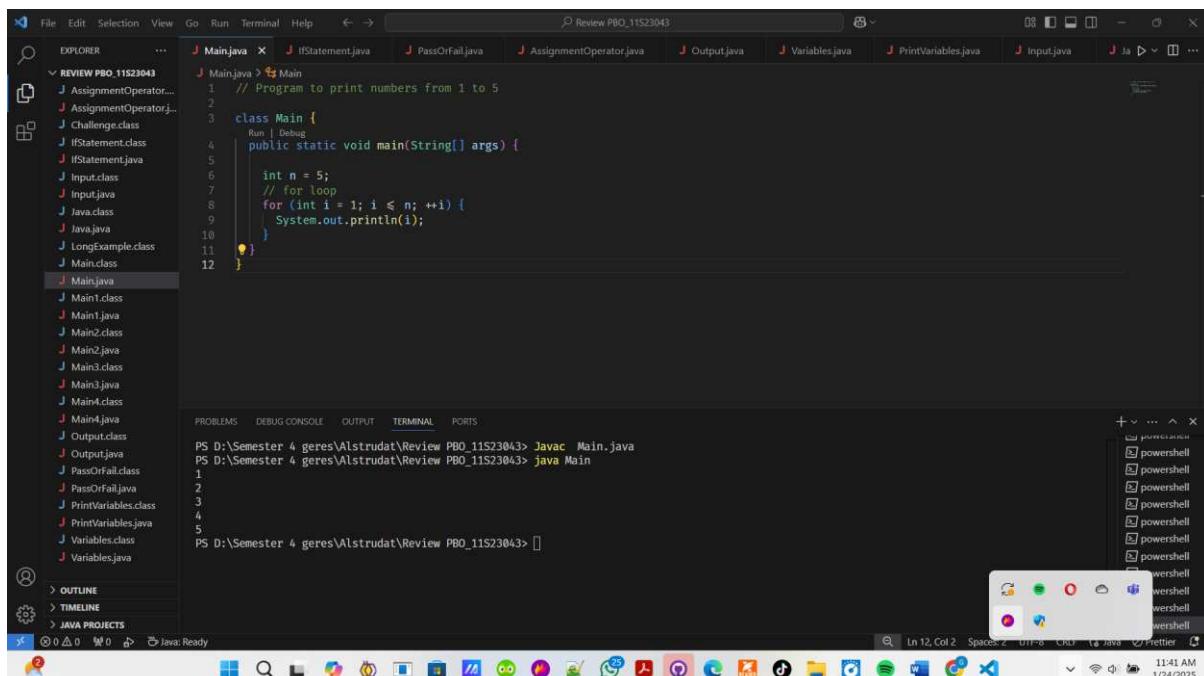


```
1 // Program to print a text 5 times
2
3 class Main {
4     public static void main(String[] args) {
5         int n = 5;
6         // for loop
7         for (int i = 1; i <= n; ++i) {
8             System.out.println("Java is fun");
9         }
10    }
11 }
12 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Java is fun
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

Display numbers from 1 to 5



```
1 // Program to print numbers from 1 to 5
2
3 class Main {
4     public static void main(String[] args) {
5         int n = 5;
6         // for loop
7         for (int i = 1; i <= n; ++i) {
8             System.out.println(i);
9         }
10    }
11 }
12 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
1
2
3
4
5
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Display Sum of n Natural Numbers

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file is open in the editor. The code defines a Main class with a main method that calculates the sum of natural numbers from 1 to 1000 using a for loop. The terminal output shows the sum is 500500.

```
class Main {
    public static void main(String[] args) {
        int sum = 0;
        int n = 1000;

        // for loop
        for (int i = 1; i <= n; ++i) {
            // body inside for loop
            sum += i; // sum = sum + i
        }

        System.out.println("Sum = " + sum);
    }
}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Sum = 500500
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file is open in the editor. The code defines a Main class with a main method that calculates the sum of natural numbers from 1 to 1000 using a for loop. The terminal output shows the sum is 500500.

```
class Main {
    public static void main(String[] args) {
        int sum = 0;
        int n = 1000;

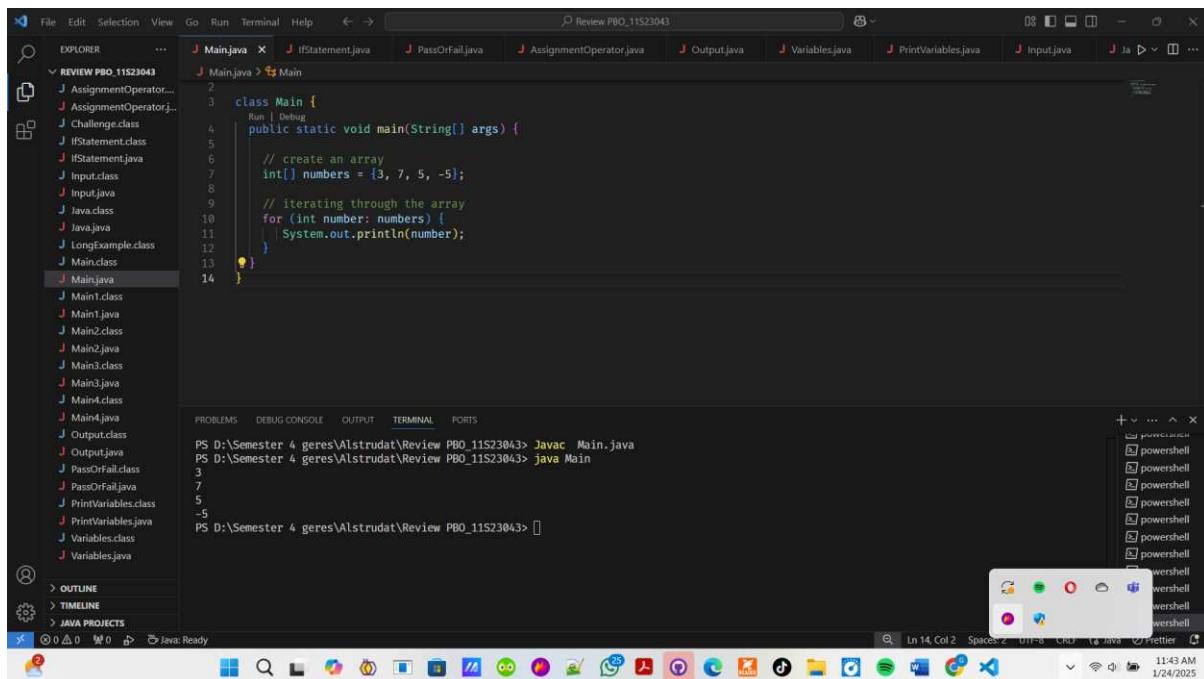
        // for loop
        for (int i = n; i >= 1; --i) {
            // body inside for loop
            sum += i; // sum = sum + i
        }

        System.out.println("Sum = " + sum);
    }
}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Sum = 500500
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java for-each Loop

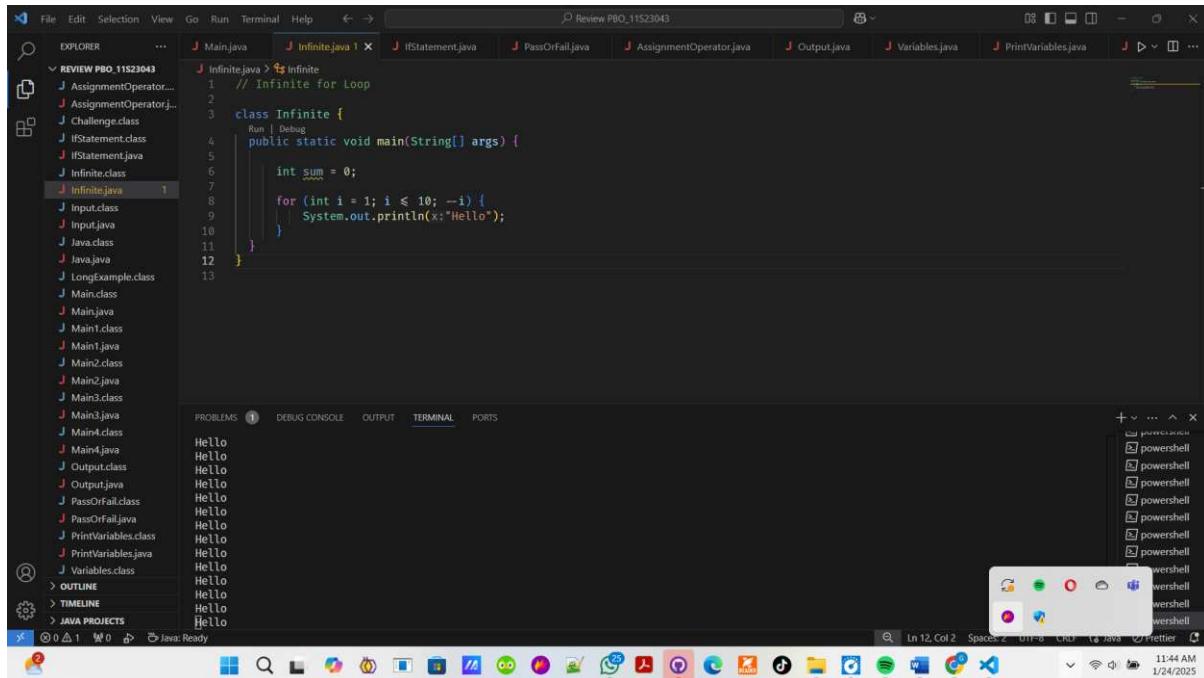


```
class Main {
    public static void main(String[] args) {
        // create an array
        int[] numbers = {3, 7, 5, -5};

        // iterating through the array
        for (int number: numbers) {
            System.out.println(number);
        }
    }
}
```

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a Java file named Main.java containing a for-each loop that iterates over an integer array named numbers, printing each element to the console. The terminal tab shows the command-line output of running the Java compiler (javac) and the Java runtime (java) on the Main class, resulting in the printed values 3, 7, 5, and -5.

Java Infinite for Loop



```
// Infinite for Loop
class Infinite {
    public static void main(String[] args) {
        int sum = 0;

        for (int i = 1; i <= 10; --i) {
            System.out.println("Hello");
        }
    }
}
```

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a Java file named Infinite.java containing an infinite for loop that prints the word "Hello" ten times to the console. The terminal tab shows the command-line output of running the Java runtime (java) on the Infinite class, resulting in ten consecutive "Hello" messages being printed.

Challenge

The screenshot shows a Java application named Main.java in the Explorer panel. The code calculates the sum of the first 5 natural numbers:

```

public class Main {
    public static int calculateSum(int n) {
        return (n * (n + 1)) / 2;
    }

    public static void main(String[] args) {
        int n = 5; // Example input
        System.out.println("The sum of the first " + n + " natural numbers is: " + calculateSum(n)); // Output: 15
    }
}

```

The terminal output shows the program running and printing the result:

```

PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The sum of the first 5 natural numbers is: 15
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>

```

✓ Java for-each Loop

Print Array Elements

The screenshot shows a Java application named Main.java in the Explorer panel. The code prints the elements of an integer array:

```

class Main {
    // print array elements
    public static void main(String[] args) {
        // create an array
        int[] numbers = {3, 9, 5, -5};

        // for each loop
        for (int number: numbers) {
            System.out.println(number);
        }
    }
}

```

The terminal output shows the program running and printing the array elements:

```

PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
3
9
5
-5
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>

```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Sum of Array Elements

The screenshot shows a Java code editor in VS Code. The code in Main.java calculates the sum of all elements in an array of integers:

```
// Calculate the sum of all elements of an array
public static void main(String[] args) {
    int[] numbers = {3, 4, 5, -5, 0, 12};
    int sum = 0;
    for (int number: numbers) {
        sum += number;
    }
    System.out.println("Sum = " + sum);
}
```

The terminal output shows the result of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main
Sum = 19
```

for loop Vs for-each loop

Using for loop

The screenshot shows a Java code editor in VS Code. The code in Main.java prints each vowel character from an array using a for loop:

```
char[] vowels = {'a', 'e', 'i', 'o', 'u'};
for (int i = 0; i < vowels.length; ++ i) {
    System.out.println(vowels[i]);
}
```

The terminal output shows the individual vowel characters:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main
a
e
i
o
u
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Using for-each Loop

The screenshot shows a Java code editor in VS Code. The file being edited is `Main.java`. The code defines a class `Main` with a `main` method. Inside the `main` method, there is a declaration of a character array `vowels` containing 'a', 'e', 'i', 'o', and 'u'. A `for` loop iterates through this array, printing each element to the console using `System.out.println`. The terminal output shows the letters 'a', 'e', 'i', 'o', and 'u' printed sequentially.

```
1 class Main {  
2     public static void main(String[] args) {  
3         char[] vowels = {'a', 'e', 'i', 'o', 'u'};  
4         // Iterating through an array using the for-each loop  
5         for (char item: vowels) {  
6             System.out.println(item);  
7         }  
8     }  
9 }  
10  
11 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main  
a  
e  
i  
o  
u  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```

Challenge

The screenshot shows a Java code editor in VS Code. The file being edited is `Main.java`. The code defines a class `Main` with a `main` method. Inside the `main` method, there is a call to a static method `calculateSum` which takes an integer array as an argument. The `calculateSum` method initializes a variable `sum` to 0 and then iterates through the array, adding each element to `sum`. Finally, it returns the total sum. The `main` method then prints the result to the console. The terminal output shows the sum of the array elements as 150.

```
1 public class Main {  
2     public static int calculateSum(int[] arr) {  
3         int sum = 0;  
4         for (int num : arr) {  
5             sum += num;  
6         }  
7         return sum;  
8     }  
9  
10    public static void main(String[] args) {  
11        int[] arr = {10, 20, 30, 40, 50}; // Example array  
12        System.out.println("The sum of the array elements is: " + calculateSum(arr)); // Output: 150  
13    }  
14 }  
15 
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main  
The sum of the array elements is: 150  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```

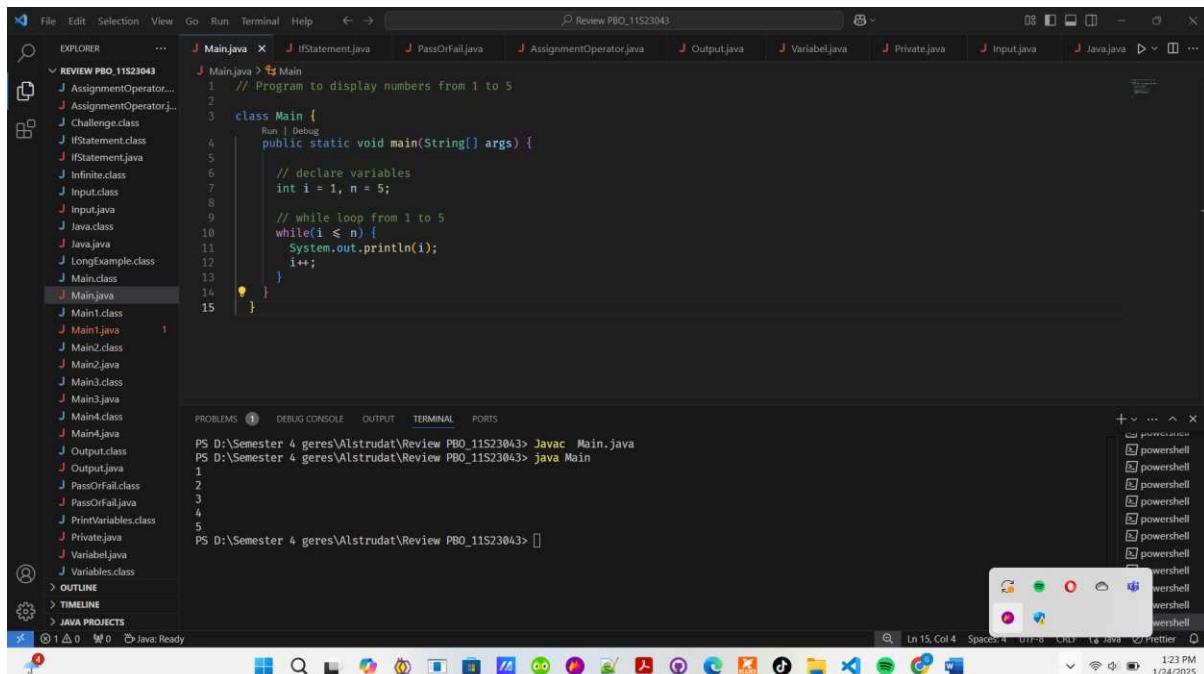
11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java while and do...while Loop

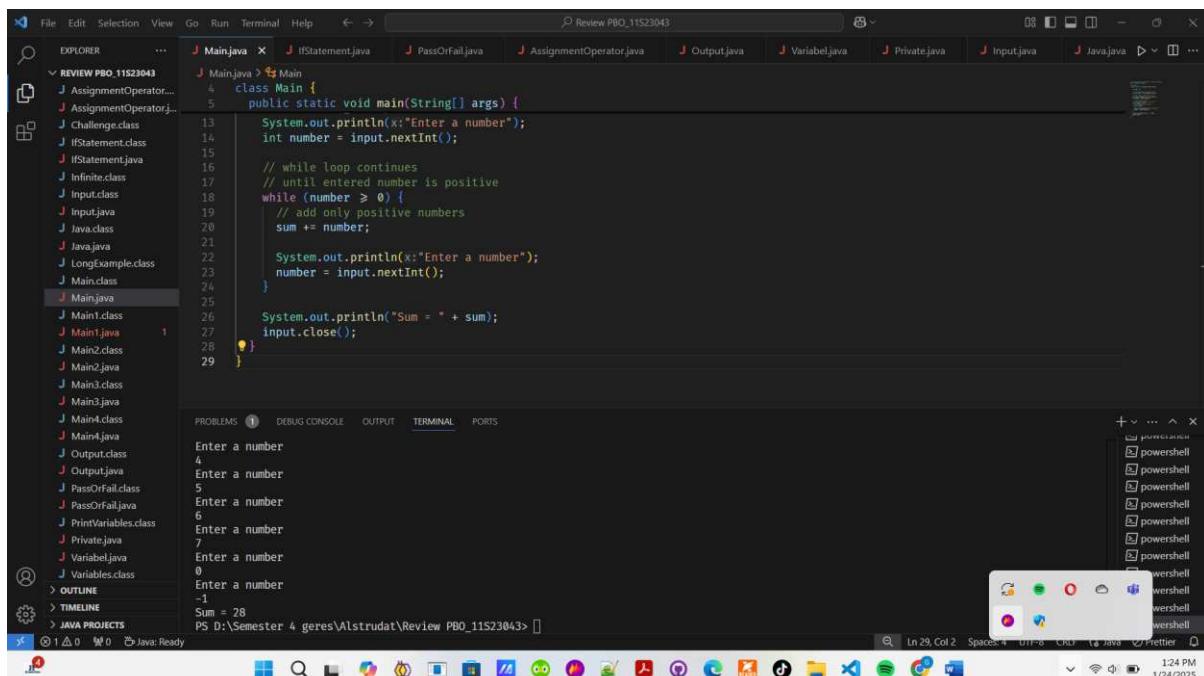
Java while loop

Display Numbers from 1 to 5



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043
EXPLORER J Main.java x J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J Private.java J Input.java J Java.java
J AssignmentOperator... J IfStatement.class J Infinite.class J Input.class J Java.class J LongExample.class J Main.class J Main.java J Main1.class J Main1.java J Main2.class J Main2.java J Main3.class J Main3.java J Main4.class J Main4.java J Output.class J Output.java J PassOrFail.class J PassOrFail.java J PrintVariables.class J Private.java J Variabel.java J Variables.class
OUTLINE
TIMELINE
JAVA PROJECTS
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
1
2
3
4
5
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
Ln 15, Col 4 Spaces: 4 011-9 CRLF ⌂ Java ⌂ Betterer ⌂
1/23 PM 1/24/2025
```

Sum of Positive Numbers Only



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043
EXPLORER J Main.java x J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J Private.java J Input.java J Java.java
J AssignmentOperator... J IfStatement.class J Infinite.class J Input.class J Java.class J LongExample.class J Main.class J Main.java J Main1.class J Main1.java J Main2.class J Main2.java J Main3.class J Main3.java J Main4.class J Main4.java J Output.class J Output.java J PassOrFail.class J PassOrFail.java J PrintVariables.class J Private.java J Variabel.java J Variables.class
OUTLINE
TIMELINE
JAVA PROJECTS
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
Enter a number
4
Enter a number
5
Enter a number
6
Enter a number
7
Enter a number
0
Enter a number
-1
Sum = 28
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
Ln 29, Col 2 Spaces: 4 011-9 CRLF ⌂ Java ⌂ Betterer ⌂
1/24 PM 1/24/2025
```

Java do...while loop

Display Numbers from 1 to 5

The screenshot shows a Java code editor interface with the following details:

- File Explorer:** On the left, a tree view shows a project named "REVIEW PBO_11S23043" containing various Java files like Main.java, Main1.java, Main2.java, etc.
- Code Editor:** The main area displays the content of Main.java:

```
1 // Java Program to display numbers from 1 to 5
2
3 import java.util.Scanner;
4
5 // Program to find the sum of natural numbers from 1 to 100.
6
7 class Main {
8     Run | Debug
9     public static void main(String[] args) {
10         int i = 1, n = 5;
11
12         // do ...while loop from 1 to 5
13         do {
14             System.out.println(i);
15             i++;
16         } while(i <= n);
17     }
18 }
```
- Terminal:** At the bottom, the terminal window shows the command "javac Main.java" followed by the output of the program which prints the numbers 1 through 5.
- System Tray:** The bottom right corner shows the Mac OS X system tray with icons for battery, signal, and volume.

Sum of Positive Numbers

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

- File Explorer:** Shows a project named "REVIEW PBO_11523043" containing various Java files: Main.java, IfStatement.java, PassOrFail.java, AssignmentOperator.java, Output.java, Variabel.java, Private.java, Input.java, and Java.java.
- Main.java Content:** The code defines a Main class with a main method that uses a Scanner to read integers from the user and calculates their sum until a non-positive number is entered.
- Terminal:** The terminal window shows the execution of the Main.java file, with the output "Sum = 3".
- System Tray:** The taskbar at the bottom includes icons for File Explorer, Task View, Taskbar settings, and system status indicators.

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Challenge

A screenshot of a Java IDE (IntelliJ IDEA) showing a Java file named Main.java. The code calculates the factorial of a non-negative integer. It includes a main method that prints the factorial of 5. The terminal output shows the result: "The factorial of 5 is: 120".

```
public static long calculateFactorial(int num) {
    if (num < 0) {
        throw new IllegalArgumentException("Number must be non-negative.");
    }
    long factorial = 1;
    for (int i = 1; i <= num; i++) {
        factorial *= i;
    }
    return factorial;
}
public static void main(String[] args) {
    // Test the function
    int num = 5;
    System.out.println("The factorial of " + num + " is: " + calculateFactorial(num));
}
```

Java break statement

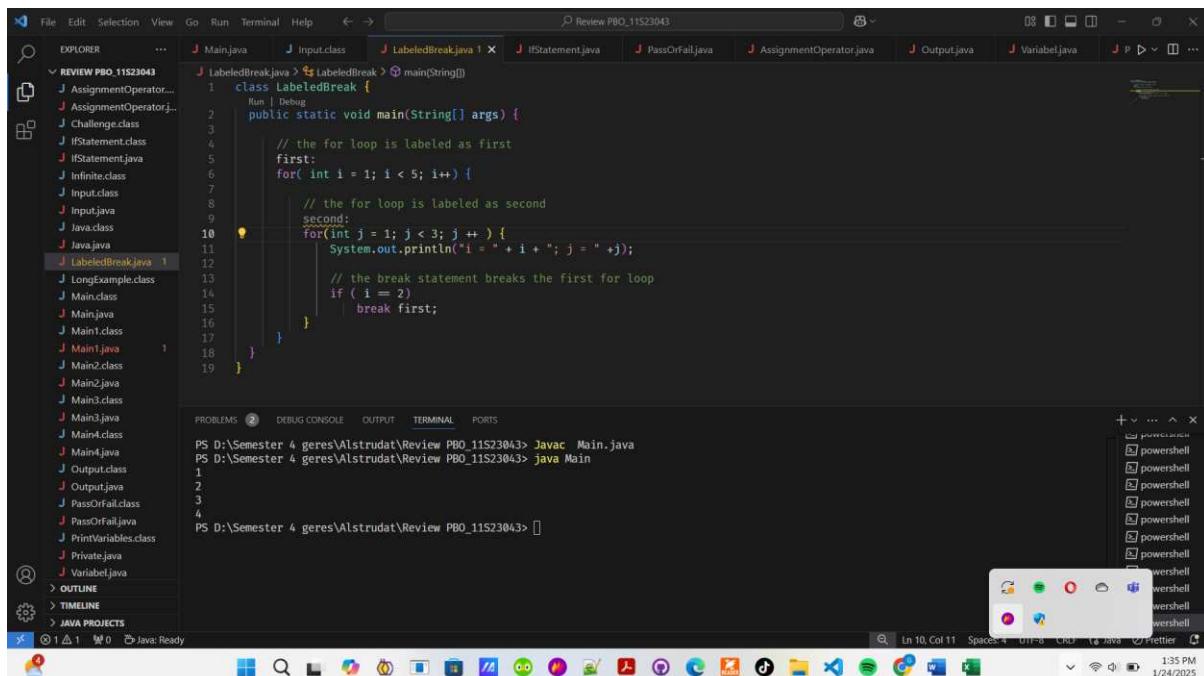
A screenshot of a Java IDE (IntelliJ IDEA) showing a Java file named Main.java. The code contains a for loop that iterates from 1 to 10. Inside the loop, it checks if the value of i is 5. If it is, the loop terminates early using a break statement. The loop then prints the value of i. The terminal output shows the values 1, 2, 3, and 4 printed.

```
public static void main(String[] args) {
    for (int i = 1; i <= 10; ++i) {
        // if the value of i is 5 the loop terminates
        if (i == 5) {
            break;
        }
        System.out.println(i);
    }
}
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

labeled break Statement



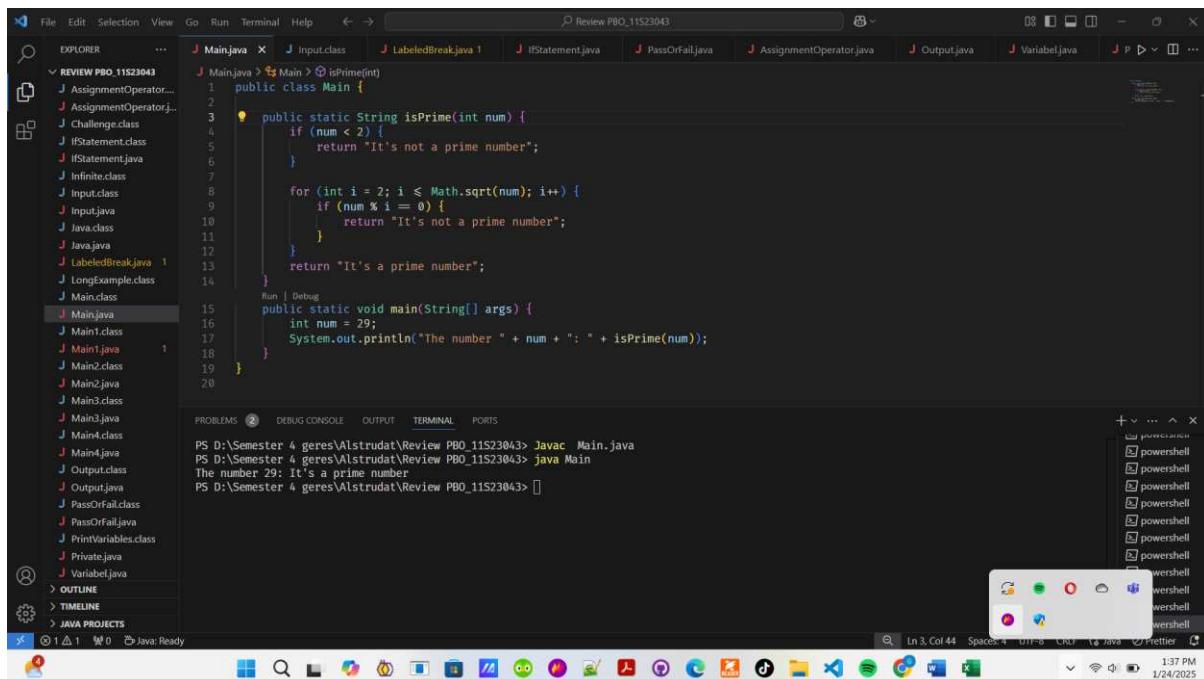
A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "Review PBO_11S23043". The left sidebar shows a file tree with several Java files. The main editor window contains the following Java code:

```
class LabeledBreak {
    public static void main(String[] args) {
        // the for loop is labeled as first
        first:
        for( int i = 1; i < 5; i++ ) {
            // the for loop is labeled as second
            second:
            for(int j = 1; j < 3; j++ ) {
                System.out.println("i = " + i + "; j = " + j);
                // the break statement breaks the first for loop
                if ( i == 2 )
                    break first;
            }
        }
    }
}
```

The code uses two nested for loops. The inner loop is labeled "second". A break statement within the inner loop breaks out of the outer loop labeled "first". The output of the code is shown in the terminal:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
1
2
3
4
```

Challenge



A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "Review PBO_11S23043". The left sidebar shows a file tree with several Java files. The main editor window contains the following Java code:

```
public class Main {
    public static String isPrime(int num) {
        if (num < 2) {
            return "It's not a prime number";
        }

        for (int i = 2; i ≤ Math.sqrt(num); i++) {
            if (num % i == 0) {
                return "It's not a prime number";
            }
        }
        return "It's a prime number";
    }

    public static void main(String[] args) {
        int num = 29;
        System.out.println("The number " + num + ": " + isPrime(num));
    }
}
```

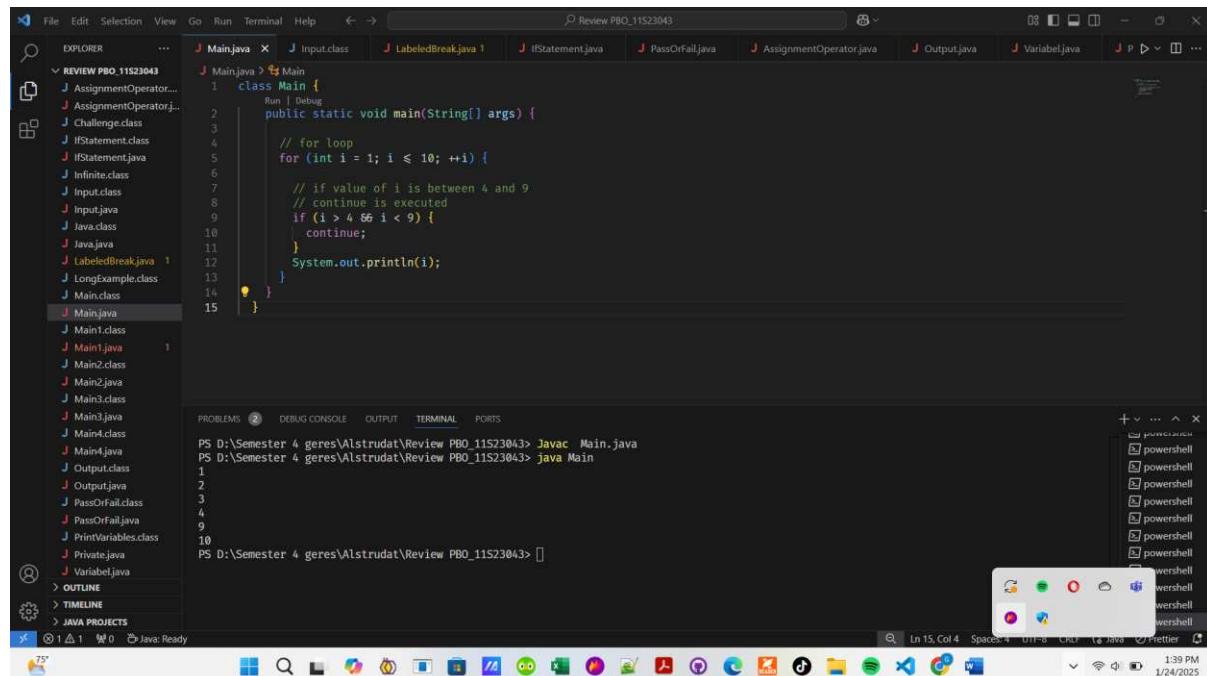
The code defines a class named Main with a static method isPrime that checks if a given number is prime. It also has a main method that prints the result of calling isPrime on the number 29. The output of the code is shown in the terminal:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
The number 29: It's a prime number
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

✓ Java continue Statement

Java continue

Java continue statement



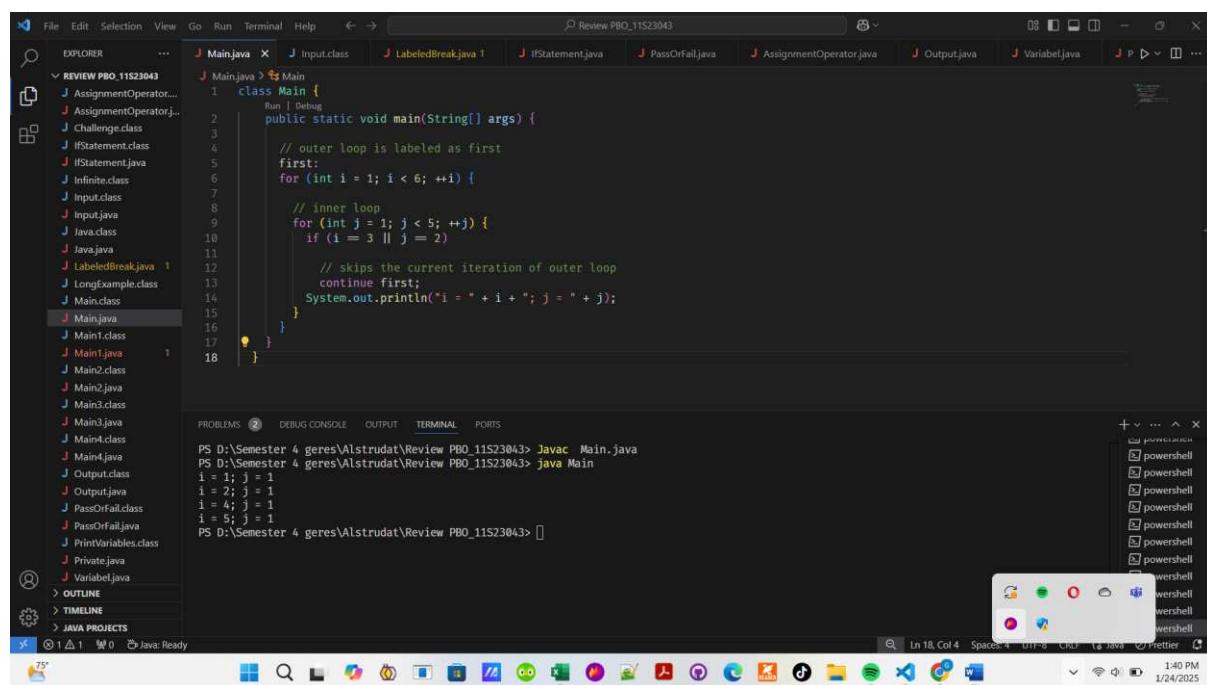
```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER ... J Main.java x J Input.class J LabeledBreak.java J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J P D ...
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3         // for loop
4         for (int i = 1; i ≤ 10; ++i) {
5             // if value of i is between 4 and 9
6             // continue is executed
7             if (i > 4 && i < 9) {
8                 continue;
9             }
10            System.out.println(i);
11        }
12    }
13}
14}
15}

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
1
2
3
4
9
10

PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

Ln 15, Col 4 Spaces: 4 011-9 CRLF ⌂ Java ⌂ prettier ⌂
11:59 PM 1/24/2025
```

labeled continue Statement



```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER ... J Main.java x J Input.class J LabeledBreak.java J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J P D ...
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3         // outer loop is labeled as first
4         first:
5         for (int i = 1; i < 6; ++i) {
6             // inner loop
7             for (int j = 1; j < 5; ++j) {
8                 if (i = 3 || j = 2) {
9                     // skips the current iteration of outer loop
10                    continue first;
11                    System.out.println("i = " + i + "; j = " + j);
12                }
13            }
14        }
15    }
16}
17}
18}

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
i = 1; j = 1
i = 2; j = 1
i = 4; j = 1
i = 5; j = 1

PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

Ln 18, Col 4 Spaces: 4 011-9 CRLF ⌂ Java ⌂ prettier ⌂
14:00 PM 1/24/2025
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java switch Statement

Java switch Statement

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

- File Explorer:** Shows files in the current project, including Main.java, Input.class, LabeledBreak.java, IfStatement.java, PassOrFail.java, AssignmentOperator.java, Output.java, Variabel.java, and several other class files.
- Code Editor:** The Main.java file is open, displaying Java code that uses a switch statement to determine the size based on a number. The code is as follows:

```
1 // Java Program to check the size
2 // using the switch ... case statement
3
4 class Main {
5     public static void main(String[] args) {
6
7         int number = 44;
8         String size;
9
10        // switch statement to check size
11        switch (number) {
12
13            case 29:
14                size = "Small";
15                break;
16
17            case 42:
18                size = "Medium";
19                break;
20
21        } // match the value of week
22    }
23}
```

- Terminal:** The terminal window shows the command-line interface for running the Java application. It displays the path D:\Semester 4\geres\Alstudat\Review PBO_11S23043, the Java command (javac Main.java), and the execution command (java Main). The output indicates the size is Large.

```
PS D:\Semester 4\geres\Alstudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstudat\Review PBO_11S23043> java Main
Size: Large
PS D:\Semester 4\geres\Alstudat\Review PBO_11S23043> [ ]
```

break Statement in Java switch...case

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

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing various Java files like Main.java, Input.class, LabeledBreak.java, IfStatement.java, PassOrFall.java, AssignmentOperator.java, Output.java, and Variabel.java.
- Code Editor:** The main editor window displays the "Main.java" file with the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         int expression = 2;  
4  
5         // switch statement to check size  
6         switch (expression) {  
7             case 1:  
8                 System.out.println(x:"Case 1");  
9  
10            // matching case  
11            case 2:  
12                System.out.println(x:"Case 2");  
13  
14            case 3:  
15                System.out.println(x:"Case 3");  
16  
17            default:  
18                System.out.println(x:"Default case");  
19  
20        }  
21    }  
22}
```
- Terminal:** The terminal window shows the command "javac Main.java" followed by the output of the program execution:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main  
Case 2  
Case 3  
Default case
```
- Status Bar:** The bottom status bar indicates "Java: Ready".

11S2215 - Algorithms and Data Structures

Laporan Praktikum



default Case in Java switch-case

The screenshot shows a Java file named Main.java in the Explorer pane. The code defines a Main class with a main method. Inside the main method, there is a switch statement with cases for 2 and 3, and a default case. The output window shows the program's execution and its result.

```
class Main {
    public static void main(String[] args) {
        int expression = 9;
        switch(expression) {
            case 2:
                System.out.println("Small Size");
                break;
            case 3:
                System.out.println("Large Size");
                break;
            default:
                System.out.println("Unknown Size");
        }
    }
}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Unknown Size
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Challenge

The screenshot shows a Java file named Main.java in the Explorer pane. The code defines a Main class with a calculate method that performs basic arithmetic operations (addition, subtraction, multiplication, division). It includes error handling for division by zero and invalid operators. The output window shows the program's execution and its result.

```
public class Main {
    // Method to perform basic arithmetic operations
    public static double calculate(double num1, char op, double num2) {
        switch (op) {
            case '+':
                return num1 + num2;
            case '-':
                return num1 - num2;
            case '*':
                return num1 * num2;
            case '/':
                if (num2 == 0) {
                    throw new ArithmeticException("Division by zero is not allowed.");
                }
                return num1 / num2;
            default:
                throw new IllegalArgumentException("Invalid operator. Use +, -, *, or /.");
        }
    }
}
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The result of 5.0 + 3.0 is: 8.0
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

• Java Arrays

Access Array Elements

```

File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER J Main.java X J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J P D ...
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3         // create an array
4         int[] age = {12, 4, 5, 2, 5};
5
6         // access each array elements
7         System.out.println("Accessing Elements of Array:");
8         System.out.println("First Element: " + age[0]);
9         System.out.println("Second Element: " + age[1]);
10        System.out.println("Third Element: " + age[2]);
11        System.out.println("Fourth Element: " + age[3]);
12        System.out.println("Fifth Element: " + age[4]);
13    }
14
15 }

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Accessing Elements of Array:
First Element: 12
Second Element: 4
Third Element: 5
Fourth Element: 2
Fifth Element: 5
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

```

Looping Through Array Elements

Using For Loop

```

File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER J Main.java X J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J AssignmentOperator.java J Output.java J Variabel.java J P D ...
J Main.java > Main > main(String[])
1 class Main {
2     public static void main(String[] args) {
3
4         // create an array
5         int[] age = {12, 4, 5};
6
7         // loop through the array
8         // using for loop
9         System.out.println("Using For Loop:");
10        for(int i = 0; i < age.length; i++) {
11            System.out.println(age[i]);
12        }
13    }
14
15 }

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Using For Loop:
12
4
5
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



Using the for-each Loop

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file contains the following code:

```
1 class Main {
2     public static void main(String[] args) {
3         // create an array
4         int[] age = {12, 4, 5};
5
6         // loop through the array
7         // using for-loop
8         System.out.println("Using for-each Loop:");
9         for(int a : age) {
10             System.out.println(a);
11         }
12     }
13 }
14 }
```

The terminal output shows the execution of the code:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Using for-each Loop:
12
4
5
```

Challenge

The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file contains the following code:

```
1 public class Main {
2     // Method to calculate the average of an array of numbers
3     public static double calculateAverage(int[] arr) {
4         // Validate if the array is empty
5         if (arr == null || arr.length == 0) {
6             throw new IllegalArgumentException("Array must not be empty.");
7         }
8
9         int sum = 0;
10        for (int num : arr) {
11            sum += num; // Calculate the sum of all numbers in the array
12        }
13
14        return (double) sum / arr.length; // Return the average
15    }
16
17    Run | Debug
18    public static void main(String[] args) {
19        // Example array
20        int[] arr = {10, 20, 30, 40};
21    }
22 }
```

The terminal output shows the execution of the code:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
The average of the array is: 25.0
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



✓ Java Multidimensional Arrays

2-dimensional Array

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays Java code for a 2D array:

```
class MultidimensionalArray {
    public static void main(String[] args) {
        // create a 2d array
        int[][] a = {
            {1, 2, 3},
            {4, 5, 6, 9},
            {7}
        };

        // calculate the length of each row
        System.out.println("Length of row 1: " + a[0].length);
        System.out.println("Length of row 2: " + a[1].length);
        System.out.println("Length of row 3: " + a[2].length);
    }
}
```

The terminal tab shows the output of running the code:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac MultidimensionalArray.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java MultidimensionalArray
Length of row 1: 3
Length of row 2: 4
Length of row 3: 1
```

The status bar at the bottom right indicates the date and time: 3:41 PM / 1/24/2025.

3-dimensional Array

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays Java code for a 3D array:

```
class ThreeArray {
    public static void main(String[] args) {
        // create a 3d array
        int[][][] test = {
            {
                {1, -2, 3},
                {2, 3, 4}
            },
            {
                {-4, -5, 6, 9},
                {1},
                {2, 3}
            }
        };

        // for.. each loop to iterate through elements of 3d array
        for (int[][] array2D: test) {
            for (int[] array1D: array2D) {
                for(int item: array1D) {
                    System.out.println(item);
                }
            }
        }
    }
}
```

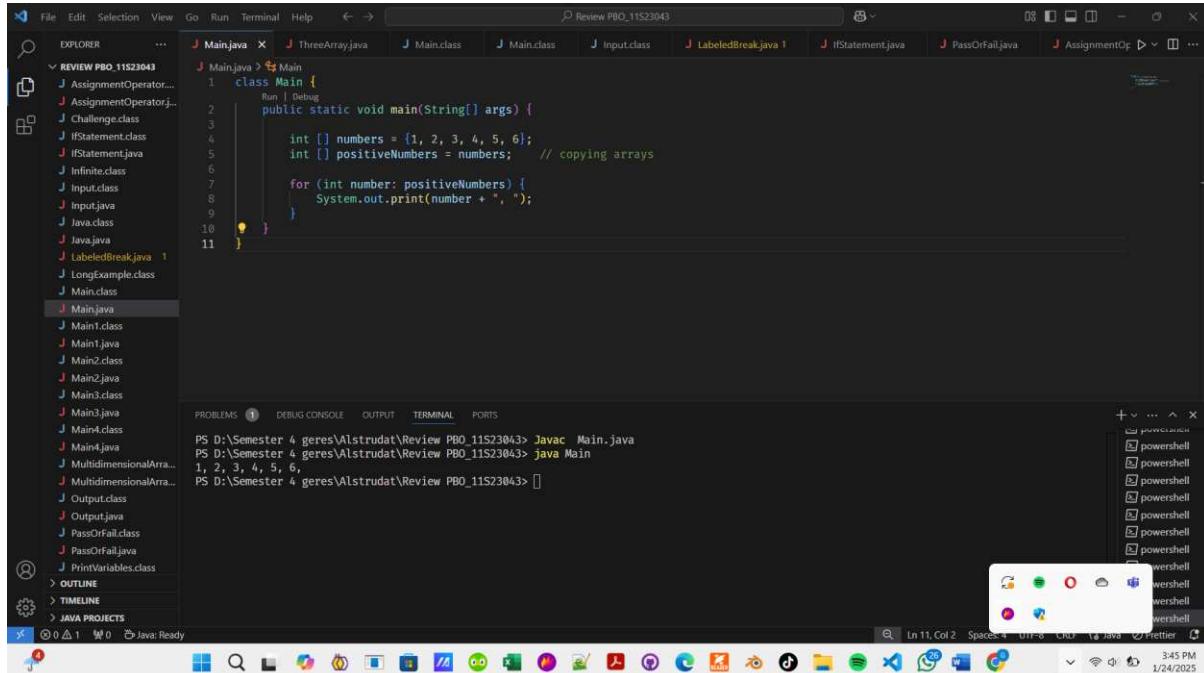
The terminal tab shows the output of running the code:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac ThreeArray.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java ThreeArray
1
-2
3
2
3
4
-4
-5
6
9
1
2
3
```

The status bar at the bottom right indicates the date and time: 3:44 PM / 1/24/2025.

✓ Java Copy Arrays

Copying Arrays Using Assignment Operator

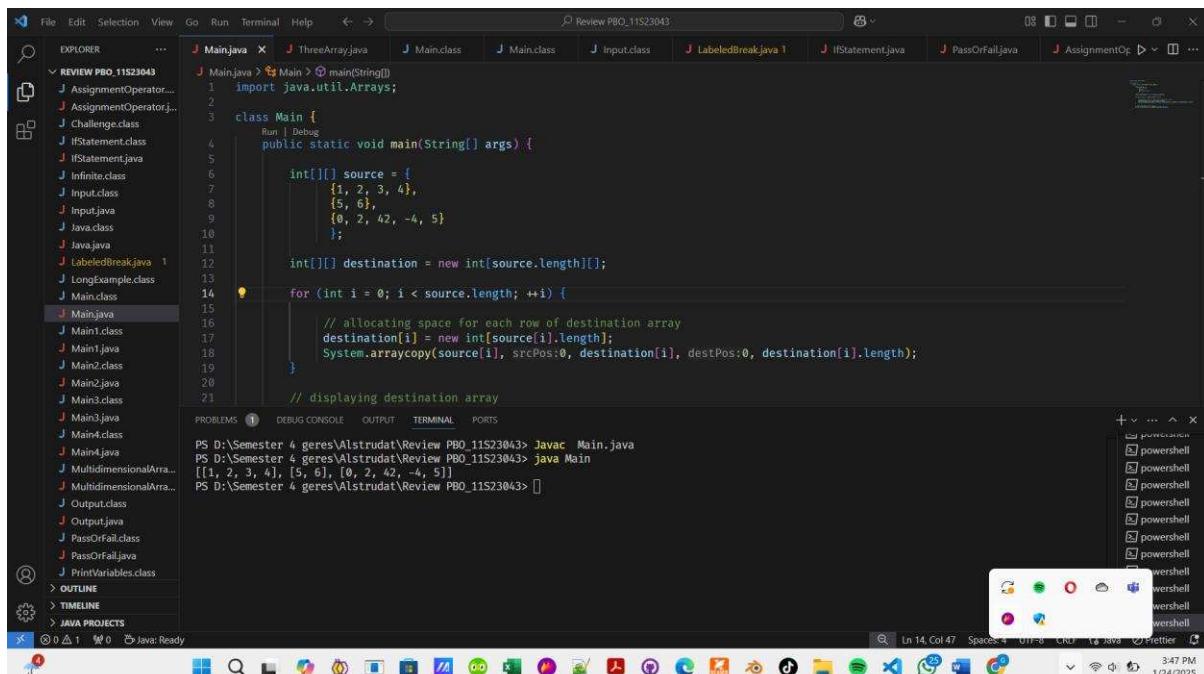


```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER ... J Main.java X J ThreeArray.java J Main.class J Main.class J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J AssignmentOp ...
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3
4         int [] numbers = {1, 2, 3, 4, 5, 6};
5         int [] positiveNumbers = numbers; // copying arrays
6
7         for (int number: positiveNumbers) {
8             System.out.print(number + " ");
9         }
10    }
11 }

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
1, 2, 3, 4, 5, 6,
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

OUTLINE TIMELINE JAVA PROJECTS
0 0 1 W Java Ready
```

Copying 2d Arrays using arraycopy()



```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER ... J Main.java X J ThreeArray.java J Main.class J Main.class J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J AssignmentOp ...
J Main.java > Main > main(String[])
1 import java.util.Arrays;
2
3 class Main {
4     public static void main(String[] args) {
5
6         int[][] source = {
7             {1, 2, 3, 4},
8             {5, 6},
9             {0, 2, 42, -4, 5}
10        };
11
12         int[][] destination = new int[source.length][];
13
14         for (int i = 0; i < source.length; ++i) {
15
16             // allocating space for each row of destination array
17             destination[i] = new int[source[i].length];
18             System.arraycopy(source[i], 0, destination[i], 0, source[i].length);
19
20         }
21
22         // displaying destination array
23     }
24 }

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
[[1, 2, 3, 4], [5, 6], [0, 2, 42, -4, 5]]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]

OUTLINE TIMELINE JAVA PROJECTS
0 0 1 W Java Ready
```

✓ Java Class and Objects

Java Class

```
class ClassName {  
    // fields  
    // methods  
}
```

Java Objects

```
className object = new className();  
  
// for Bicycle class  
Bicycle sportsBicycle = new Bicycle();  
  
Bicycle touringBicycle = new Bicycle();
```

Create objects inside the same class

The screenshot shows the Eclipse IDE interface with the following details:

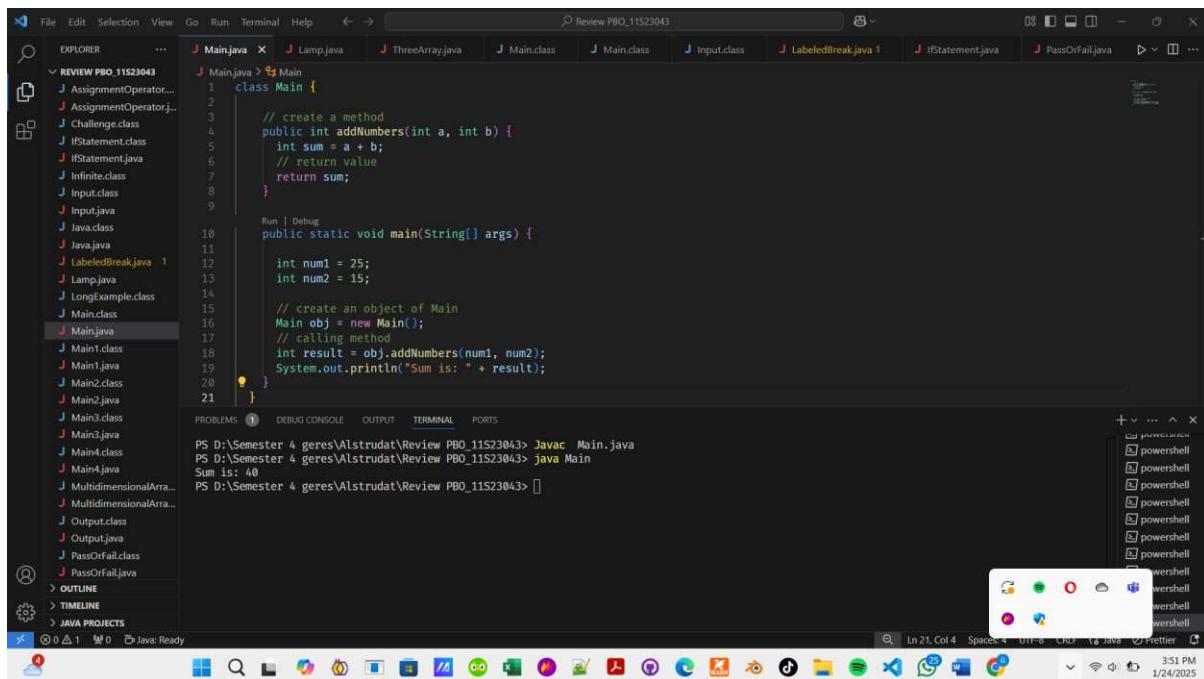
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Review PBO_11523043.
- Left Sidebar (Project Explorer):** Shows files like Main.java, Lamp.java, ThreeArray.java, Main.class, Main.class, Input.class, LabeledBreak.java, IfStatement.java, and PassOrFail.java.
- Central Area (Editor):** Lamp.java code:

```
1 class Lamp {  
2     // stores the value for light  
3     // true if light is on  
4     // false if light is off  
5     boolean isOn;  
6  
7     // method to turn on the light  
8     void turnOn() {  
9         isOn = true;  
10        System.out.println("Light on? " + isOn);  
11    }  
12  
13 }  
14  
Run | Debug  
public static void main(String[] args) {  
15  
16     // create an object of Lamp  
17     Lamp led = new Lamp();  
18  
19     // access method using object  
20     led.turnOn();  
21 }
```
- Bottom Bar (Terminal):** PS D:\Semester 4\geres\Alstrudat\Review PBO_11523043> Javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11523043> java Main
[[1, 2, 3, 4], [5, 6], [0, 2, 42, -4, 5]]
PS D:\Semester 4\geres\Alstrudat\Review PBO_11523043>
- Right Sidebar (Recent Editors):** Powershell (multiple entries).
- Bottom Icons:** Taskbar icons for various applications like File Explorer, Task Manager, and Network.

11S2215 - Algorithms and Data Structures

Laporan Praktikum

Java Methods



The screenshot shows a Java project in VS Code with several files listed in the Explorer sidebar. The main file, Main.java, is open in the editor. It contains the following code:

```
class Main {
    // create a method
    public int addNumbers(int a, int b) {
        int sum = a + b;
        // return value
        return sum;
    }

    Run | Debug
    public static void main(String[] args) {
        int num1 = 25;
        int num2 = 15;

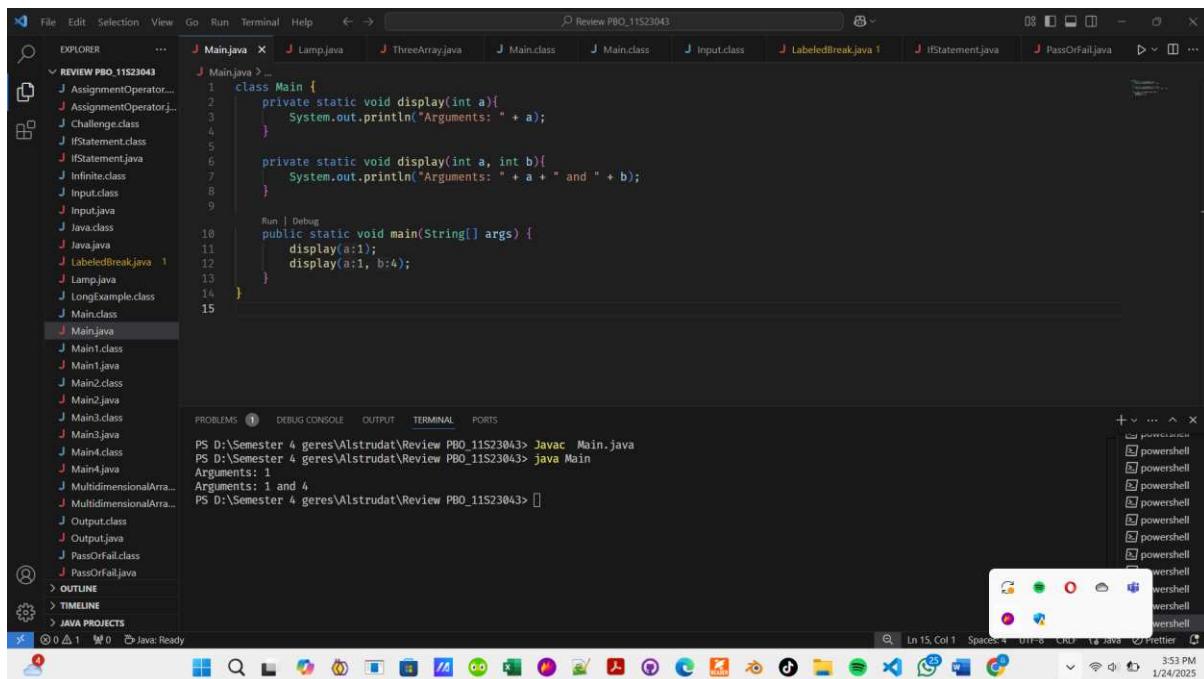
        // create an object of Main
        Main obj = new Main();
        // calling method
        int result = obj.addNumbers(num1, num2);
        System.out.println("Sum is: " + result);
    }
}
```

The terminal below shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Sum is: 40
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

✓ Java Method Overloading

Overloading by changing the number of parameters



The screenshot shows a Java project in VS Code with several files listed in the Explorer sidebar. The main file, Main.java, is open in the editor. It contains the following code:

```
class Main {
    private static void display(int a){
        System.out.println("Arguments: " + a);
    }

    private static void display(int a, int b){
        System.out.println("Arguments: " + a + " and " + b);
    }

    Run | Debug
    public static void main(String[] args) {
        display(a:1);
        display(a:1, b:4);
    }
}
```

The terminal below shows the output of running the code:

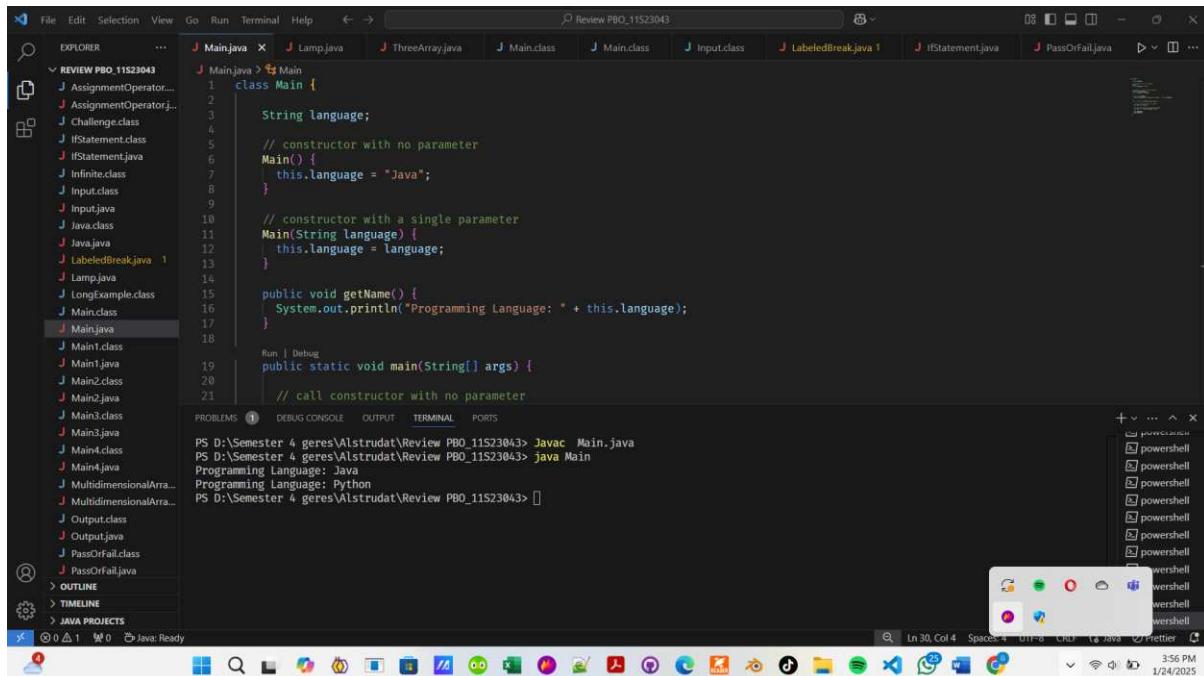
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Arguments: 1
Arguments: 1 and 4
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java Constructors

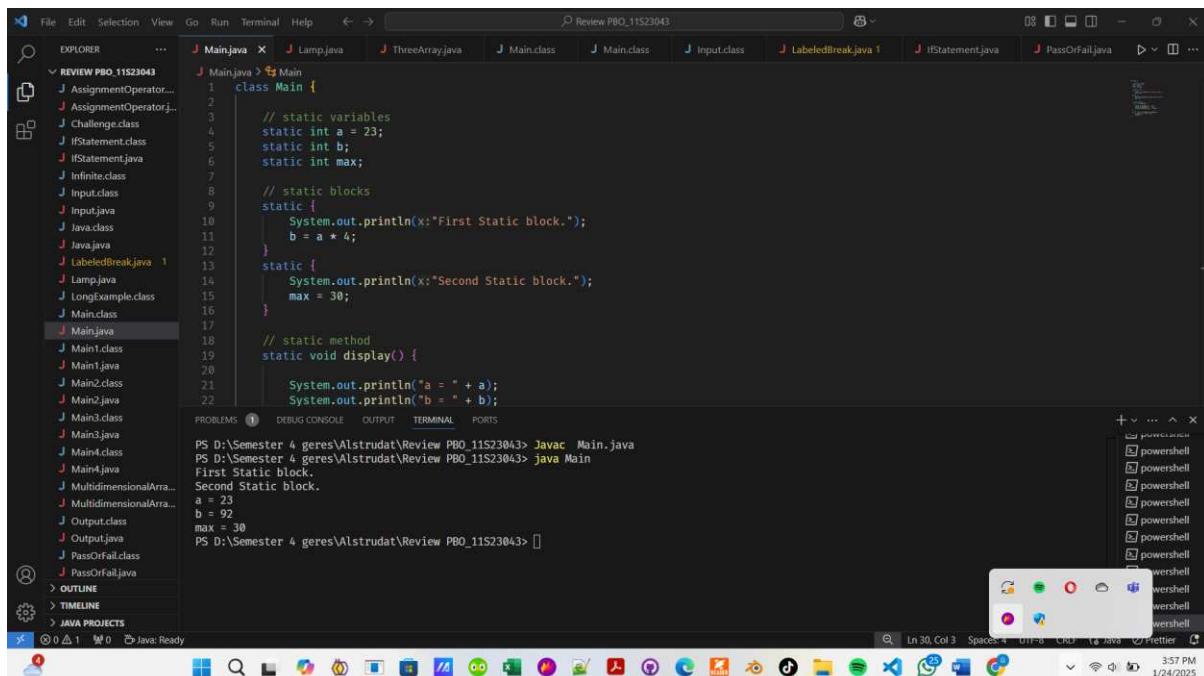
Java Constructor Overloading



```
File Edit Selection View Go Run Terminal Help ← → Review PBO_11S23043
EXPLORER ... J Main.java X J Lamp.java J ThreeArray.java J Main.class J Main.class J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java
J Main.java > Main
1 class Main {
2
3     String language;
4
5     // constructor with no parameter
6     Main() {
7         this.language = "Java";
8     }
9
10    // constructor with a single parameter
11    Main(String language) {
12        this.language = language;
13    }
14
15    public void getName() {
16        System.out.println("Programming Language: " + this.language);
17    }
18
19    public static void main(String[] args) {
20
21        // call constructor with no parameter
22
23        Run | Debug
24        public static void main(String[] args) {
25
26            Main.main();
27
28        } // call constructor with no parameter
29
30        Main.main();
31
32        Main.main();
33
34        Main.main();
35
36        Main.main();
37
38        Main.main();
39
40        Main.main();
41
42        Main.main();
43
44        Main.main();
45
46        Main.main();
47
48        Main.main();
49
50        Main.main();
51
52        Main.main();
53
54        Main.main();
55
56        Main.main();
57
58        Main.main();
59
60        Main.main();
61
62        Main.main();
63
64        Main.main();
65
66        Main.main();
67
68        Main.main();
69
70        Main.main();
71
72        Main.main();
73
74        Main.main();
75
76        Main.main();
77
78        Main.main();
79
80        Main.main();
81
82        Main.main();
83
84        Main.main();
85
86        Main.main();
87
88        Main.main();
89
90        Main.main();
91
92        Main.main();
93
94        Main.main();
95
96        Main.main();
97
98        Main.main();
99
100       Main.main();
101
102       Main.main();
103
104       Main.main();
105
106       Main.main();
107
108       Main.main();
109
110      Main.main();
111
112      Main.main();
113
114      Main.main();
115
116      Main.main();
117
118      Main.main();
119
120      Main.main();
121
122      Main.main();
123
124      Main.main();
125
126      Main.main();
127
128      Main.main();
129
130      Main.main();
131
132      Main.main();
133
134      Main.main();
135
136      Main.main();
137
138      Main.main();
139
140      Main.main();
141
142      Main.main();
143
144      Main.main();
145
146      Main.main();
147
148      Main.main();
149
150      Main.main();
151
152      Main.main();
153
154      Main.main();
155
156      Main.main();
157
158      Main.main();
159
160      Main.main();
161
162      Main.main();
163
164      Main.main();
165
166      Main.main();
167
168      Main.main();
169
169
PROBLEMS ① DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Programming Language: Java
Programming Language: Python
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]
[REDACTED]
```

✓ Java Static Keyword

Use of static block in java



```
File Edit Selection View Go Run Terminal Help ← → Review PBO_11S23043
EXPLORER ... J Main.java X J Lamp.java J ThreeArray.java J Main.class J Main.class J Input.class J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java
J Main.java > Main
1 class Main {
2
3     // static variables
4     static int a = 23;
5     static int b;
6     static int max;
7
8     // static blocks
9     static {
10         System.out.println("First Static block.");
11         b = a * 4;
12     }
13
14     static {
15         System.out.println("Second Static block.");
16         max = 30;
17     }
18
19     // static method
20     static void display() {
21
22         System.out.println("a = " + a);
23         System.out.println("b = " + b);
24     }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
169
PROBLEMS ① DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> Javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
First Static block.
Second Static block.
a = 23
b = 92
max = 30
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [REDACTED]
[REDACTED]
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



✓ Java Strings

Create a String in Java

The screenshot shows a Java code editor with the following code:

```
public static void main(String[] args) {
    // create strings
    String first = "Java";
    String second = "Python";
    String third = "JavaScript";
    // print strings
    System.out.println(first); // print Java
    System.out.println(second); // print Python
    System.out.println(third); // print JavaScript
}
```

The code is run in the terminal, and the output is:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Java
Python
JavaScript
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

Challenge

The screenshot shows a Java code editor with the following code:

```
public class Main {
    // Method to check if a string is empty
    public static String isEmpty(String str) {
        if (str == null || str.isEmpty()) {
            return "String is empty";
        } else {
            return "String isn't empty";
        }
    }
    public static void main(String[] args) {
        // Example test cases
        String str1 = "Hello World!";
        String str2 = "";
        String str3 = null;
        System.out.println("Test 1: " + isEmpty(str1)); // String isn't empty
        System.out.println("Test 2: " + isEmpty(str2)); // String is empty
        System.out.println("Test 3: " + isEmpty(str3)); // String is empty
    }
}
```

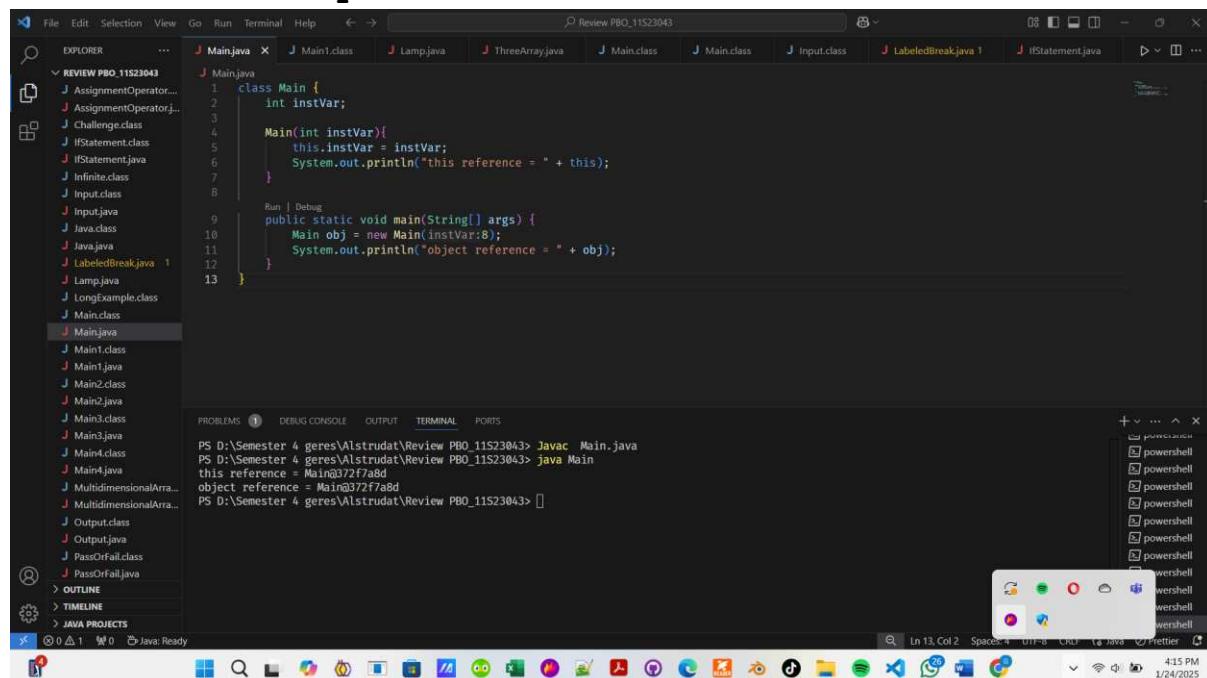
The code is run in the terminal, and the output is:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Test 1: String isn't empty
Test 2: String is empty
Test 3: String is empty
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java this Keyword



```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
```

EXPLORER ... J Main.java x J Main1.class J Lamp.java J ThreeArray.java J Main.class J Main.class J Input.class J Input.java J Java-class J Java.java J LabeledBreak.java 1 J Lamp.java J LongExample.class J Main.class J Main.java J Main1.class J Main.java J Main2.class J Main.java J Main3.class J Main.java J Main4.class J Main.java J MultidimensionalArra... J MultidimensionalArra... J Output.class J Output.java J PassOrFail.class J PassOrFail.java

```
J Main.java
1 class Main {
2     int instVar;
3
4     Main(int instVar){
5         this.instVar = instVar;
6         System.out.println("this reference = " + this);
7     }
8
9     public static void main(String[] args) {
10        Main obj = new Main(instVar:8);
11        System.out.println("object reference = " + obj);
12    }
13 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

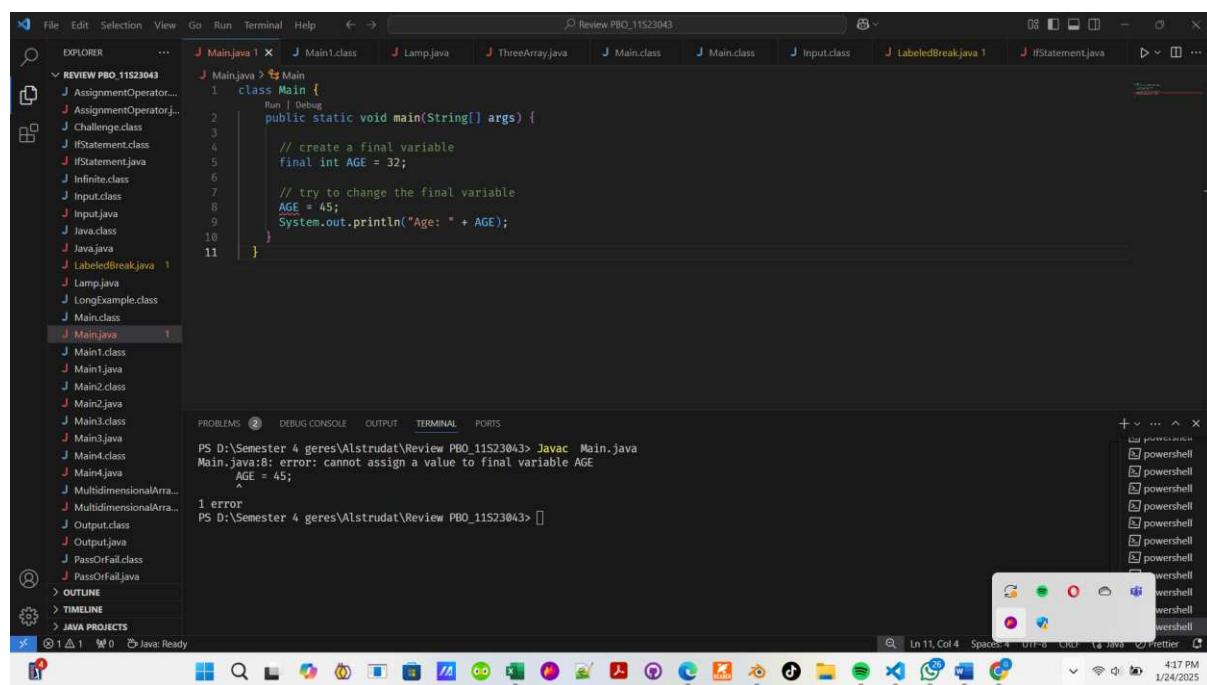
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
this reference = Main@372f7a8d
object reference = Main@372f7a8d
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]
```

Icons: File, Save, Open, Find, Copy, Paste, Cut, Undo, Redo, Run, Stop, Refresh, Terminal, Task Manager, Help, Exit.

Bottom status bar: In 13, Col 2 | Space: 4 | 011-5 | CWD | Java | Prettier | 4:15 PM | 1/24/2025

✓ Java final keyword

Java final Variable



```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
```

EXPLORER ... J Main.java 1 x J Main1.class J Lamp.java J ThreeArray.java J Main.class J Main.class J Input.class J Input.java J Java-class J Java.java J LabeledBreak.java 1 J Lamp.java J LongExample.class J Main.class J Main.java J Main1.class J Main.java J Main2.class J Main.java J Main3.class J Main.java J Main4.class J Main.java J MultidimensionalArra... J MultidimensionalArra... J Output.class J Output.java J PassOrFail.class J PassOrFail.java

```
J Main.java 1 > Main
1 class Main {
2     public static void main(String[] args) {
3
4         // create a final variable
5         final int AGE = 32;
6
7         // try to change the final variable
8         AGE = 45;
9         System.out.println("Age: " + AGE);
10    }
11 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
Main.java:8: error: cannot assign a value to final variable AGE
          AGE = 45;
                  ^
1 error
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]
```

Icons: File, Save, Open, Find, Copy, Paste, Cut, Undo, Redo, Run, Stop, Refresh, Terminal, Task Manager, Help, Exit.

Bottom status bar: In 11, Col 4 | Space: 4 | 011-5 | CWD | Java | Prettier | 4:17 PM | 1/24/2025

11S2215 - Algorithms and Data Structures

Laporan Praktikum



✓ Java Recursion

Factorial of a Number Using Recursion

The screenshot shows a Java file named Main.java in the VS Code editor. The code defines a class Main with a static method factorial that calculates the factorial of a number using recursion. The terminal output shows the execution of the program and the result of factorial(4).

```
1  static int factorial( int n ) {
2      if (n != 0) // termination condition
3          return n * factorial(n-1); // recursive call
4      else
5          return 1;
6  }
7
8  public static void main(String[] args) {
9      int number = 4, result;
10     result = factorial(number);
11     System.out.println(number + " factorial = " + result);
12 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> JavaC Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
4 factorial = 24
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

Challenge

The screenshot shows a Java file named Main.java in the VS Code editor. The code defines a class Main with a static method calculateFactorial that calculates the factorial of a number using iteration. The terminal output shows the execution of the program and the result of calculateFactorial(5).

```
1  public class Main {
2
3      // Method to calculate the factorial of a number
4      public static long calculateFactorial(int num) {
5          if (num < 0) {
6              throw new IllegalArgumentException("Number must be non-negative.");
7          }
8          long factorial = 1;
9          for (int i = 1; i <= num; i++) {
10              factorial *= i; // Multiply factorial by the current number
11          }
12          return factorial;
13      }
14
15  public static void main(String[] args) {
16      // Example usage
17      int num = 5; // Test input
18      try {
19          System.out.println("The factorial of " + num + " is: " + calculateFactorial(num));
20      } catch (Exception e) {
21          System.out.println(e.getMessage());
22      }
23  }
24 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> JavaC Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
The factorial of 5 is: 120
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

• Java OOP (II)

Java Inheritance

```

class Animal {
    String name;
    public void eat() {
        System.out.println("I can eat");
    }
}

class Dog extends Animal {
    public void display() {
        System.out.println("My name is " + name);
    }
}

class Main {
    public static void main(String[] args) {
        Run | Debug
    }
}

```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
My name is Rohu
I can eat
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>

Method Overriding

```

class Animal {
    public void displayInfo() {
        System.out.println("I am an animal.");
    }
}

class Dog extends Animal {
    @Override
    public void displayInfo() {
        System.out.println("I am a dog.");
    }
}

class Main {
    public static void main(String[] args) {
        Run | Debug
        Dog d1 = new Dog();
        d1.displayInfo();
    }
}

```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
I am a dog.
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java super

The screenshot shows the Eclipse IDE interface with the following details:

- File Explorer (left):** Shows a tree view of Java files under the project "REVIEW PBO_11S23043". The "Main.java" file is currently selected.
- Editor (center):** Displays the code for the "Main.java" file. It contains three classes: "Animal", "Dog", and "Main". The "display()" method is overridden in both "Animal" and "Dog". The "printMessage()" method is defined in "Main".

```
class Animal {
    public void display(){
        System.out.println("I am an animal");
    }
}

class Dog extends Animal {
    @Override
    public void display(){
        System.out.println("I am a dog");
    }

    public void printMessage(){
        display();
    }
}

class Main {
```
- Terminal (bottom):** Shows the command-line output of running the "javac Main.java" and "java Main" commands. The output indicates that the program runs successfully, printing "I am a dog".

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
I am a dog
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows the status "Java: Ready".
- System Tray (bottom right):** Shows icons for power shell, task manager, and other system utilities.

✓ Java Abstract Class and Abstract Methods

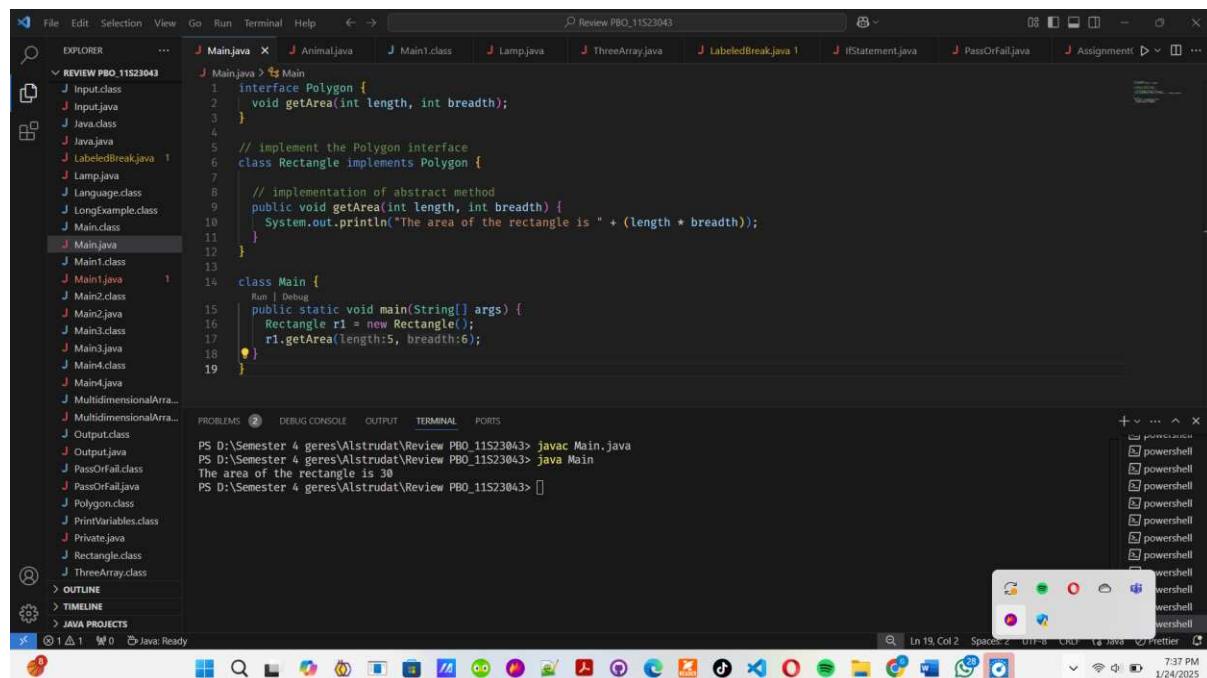
The screenshot shows a Java IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Review PBO_11S23043.
- Left Sidebar (EXPLORER):** Shows a tree view of files under REVIEW PBO_11S23043, including Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and Assignments.
- Central Area:** Code editor for Main.java. The code defines an abstract class Language with a display() method, and a Main class that extends Language and overrides the display() method to print "This is Java Programming". A code completion dropdown is visible over the word "Main" in the first line of the Main class definition.
- Bottom Navigation:** PROBLEMS, DEBUG CONSOLE, OUTPUT, TERMINAL, PORTS.
- TERMINAL:** Shows command-line output for running Main.java and Main class.
- Bottom Status Bar:** Ln 20, Col 2, Space: 2, UFT-8, CR/LF, Java Ready.
- Taskbar:** Shows icons for various applications like File Explorer, Task Manager, and browser.

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java Interface



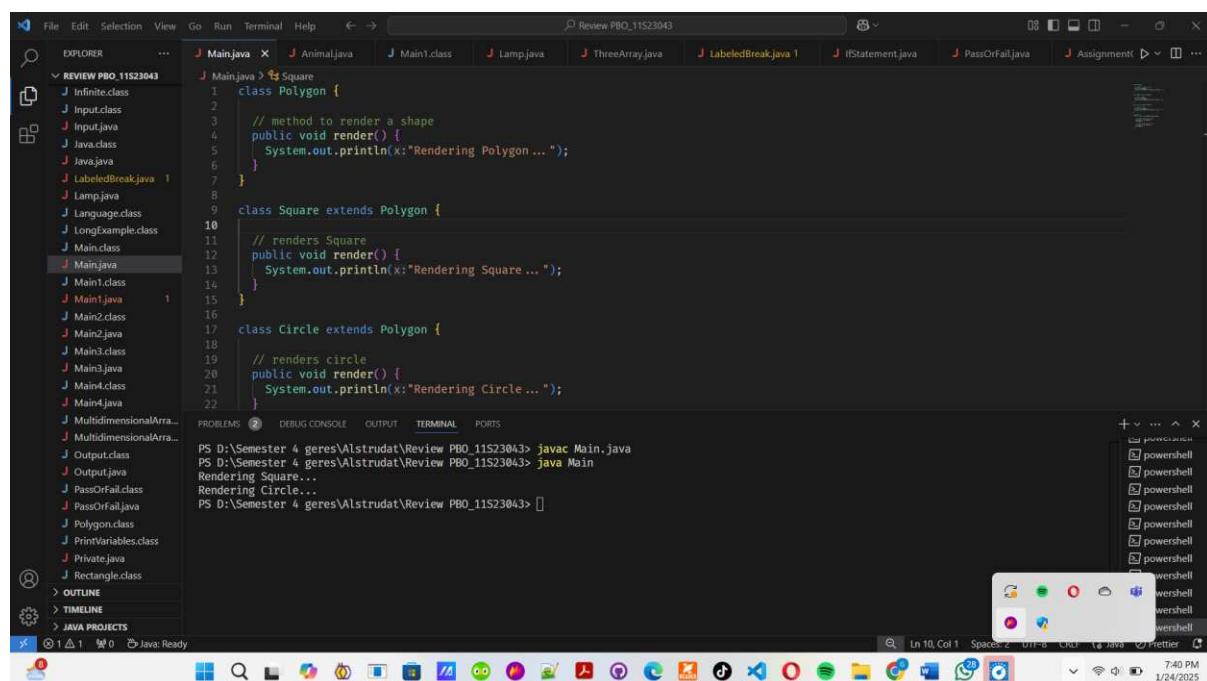
```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043

REVIEW PBO_11S23043
J Input.class J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D ... 

J Main.java > Main
1 interface Polygon {
2     void getArea(int length, int breadth);
3 }
4
5 // implement the Polygon interface
6 class Rectangle implements Polygon {
7
8     // implementation of abstract method
9     public void getArea(int length, int breadth) {
10         System.out.println("The area of the rectangle is " + (length * breadth));
11     }
12 }
13
14 class Main {
15     Run | Debug
16     public static void main(String[] args) {
17         Rectangle r1 = new Rectangle();
18         r1.getArea(length:5, breadth:6);
19     }
}

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The area of the rectangle is 30
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]
```

✓ Java Polymorphism



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043

REVIEW PBO_11S23043
J Infinite.class J Input.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D ... 

J Main.java > Square
1 class Polygon {
2
3     // method to render a shape
4     public void render() {
5         System.out.println(x:"Rendering Polygon ... ");
6     }
7 }
8
9 class Square extends Polygon {
10
11     // renders Square
12     public void render() {
13         System.out.println(x:"Rendering Square ... ");
14     }
15 }
16
17 class Circle extends Polygon {
18
19     // renders circle
20     public void render() {
21         System.out.println(x:"Rendering Circle... ");
22     }
}

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Rendering Square...
Rendering Circle...
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



✓ Java Encapsulation

The screenshot shows the VS Code interface with the title bar "Review PBO_11S23043". The Explorer sidebar lists files like Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and AssignmentK. The main editor window contains the following Java code:

```
class Area {
    // fields to calculate area
    int length;
    int breadth;

    // constructor to initialize values
    Area(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    // method to calculate area
    public void getArea() {
        int area = length * breadth;
        System.out.println("Area: " + area);
    }
}

class Main {
    public static void main(String[] args) {
        Run | Debug
    }
}
```

The terminal output shows the execution of Main.java and the output "Area: 30". The status bar at the bottom right indicates "7:42 PM" and "1/24/2025".

• Java OOP (III)

✓ Java Nested and Inner Class

The screenshot shows the VS Code interface with the title bar "Review PBO_11S23043". The Explorer sidebar lists files like Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and AssignmentK. The main editor window contains the following Java code:

```
class CPU {
    double price;
    // nested class
    class Processor{
        // members of nested class
        double cores;
        String manufacturer;

        double getCache(){
            return 4.3;
        }
    }

    // nested protected class
    protected class RAM{
        // members of protected nested class
        double memory;
        String manufacturer;

        double getClockSpeed(){
            return 5.5;
        }
    }
}

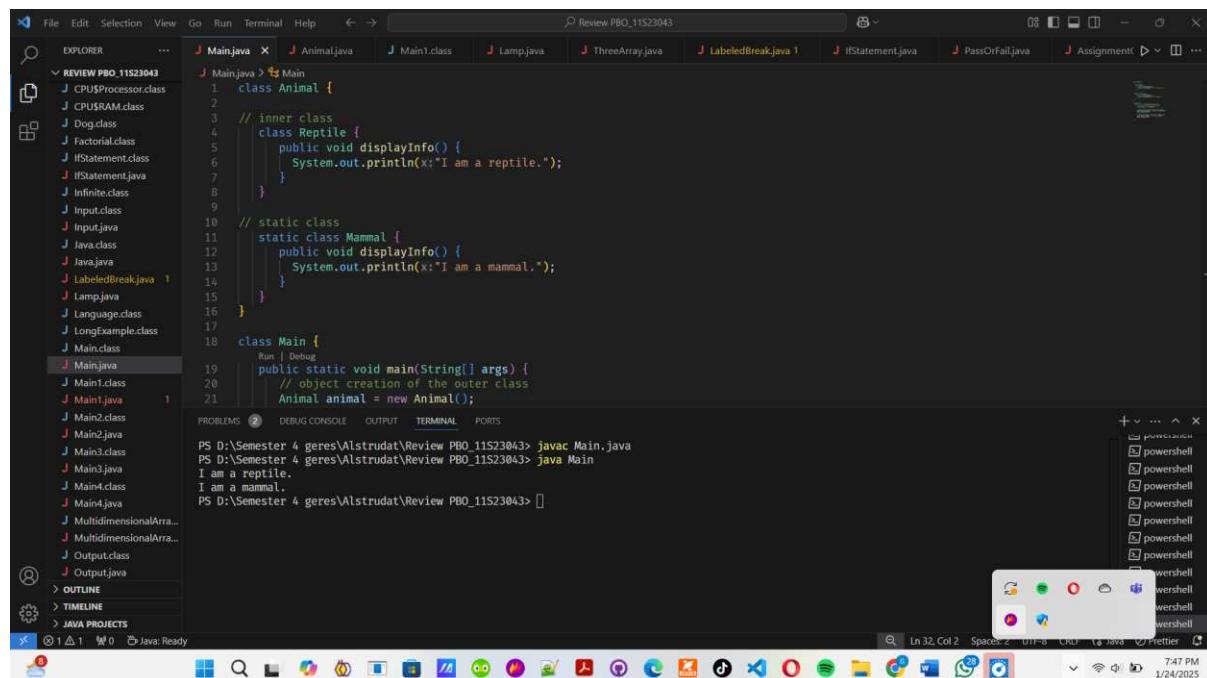
Processor Cache = 4.3
Ram Clock speed = 5.5
```

The terminal output shows the execution of Main.java and the output "Processor Cache = 4.3" and "Ram Clock speed = 5.5". The status bar at the bottom right indicates "7:45 PM" and "1/24/2025".

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java Nested Static Class



The screenshot shows the Visual Studio Code interface with the following details:

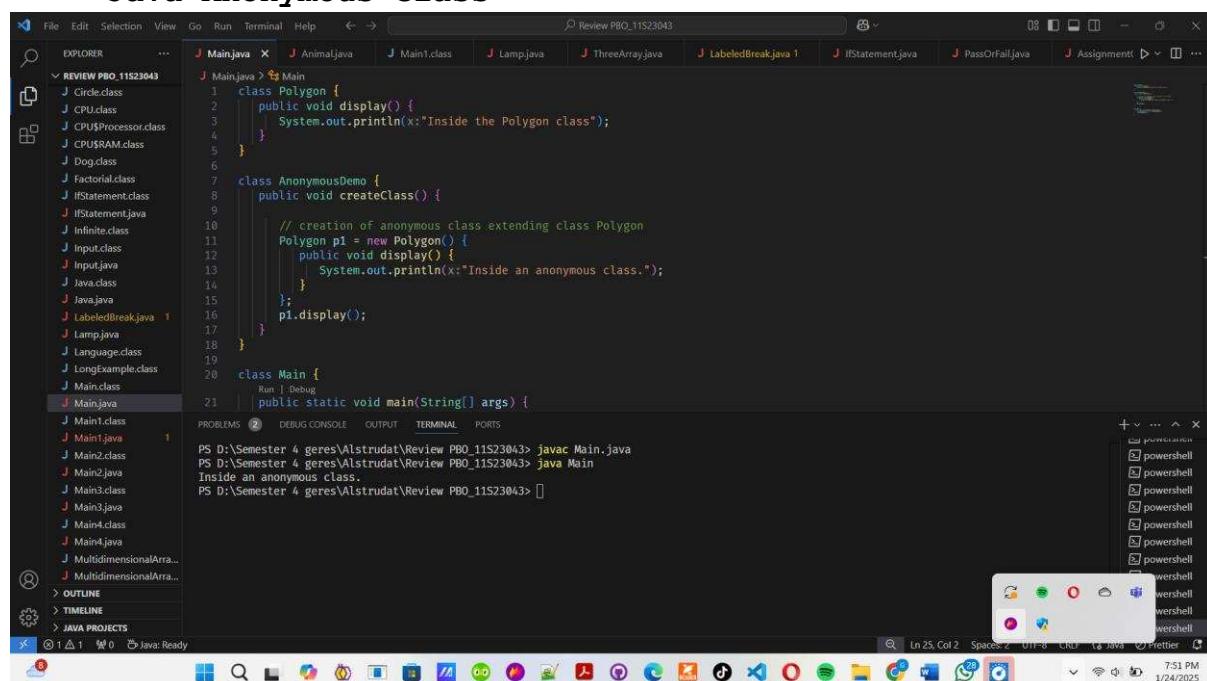
- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing numerous Java files.
- Code Editor:** Displays the following Java code:

```
class Main {
    class Animal {
        // inner class
        class Reptile {
            public void displayInfo() {
                System.out.println("I am a reptile.");
            }
        }
        // static class
        static class Mammal {
            public void displayInfo() {
                System.out.println("I am a mammal.");
            }
        }
    }
    class Main {
        public static void main(String[] args) {
            // object creation of the outer class
            Animal animal = new Animal();
        }
    }
}
```

The terminal output shows the execution of the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
I am a reptile.
I am a mammal.
```

✓ Java Anonymous Class



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing numerous Java files.
- Code Editor:** Displays the following Java code:

```
class Main {
    class Polygon {
        public void display() {
            System.out.println("Inside the Polygon class");
        }
    }
    class AnonymousDemo {
        public void createClass() {
            // creation of anonymous class extending class Polygon
            Polygon p1 = new Polygon() {
                public void display() {
                    System.out.println("Inside an anonymous class.");
                }
            };
            p1.display();
        }
    }
    class Main {
        public static void main(String[] args) {
        }
    }
}
```

The terminal output shows the execution of the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Inside an anonymous class.
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum



✓ Java Singleton Class

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it shows a tree view of files under the project "REVIEW PRO_11S23043". The "Main.java" file is currently selected.
- Code Editor:** The main area displays the Java code for "Main.java". The code implements a Database singleton pattern.

```
class Database {
    private static Database dbObject;

    private Database() {}

    public static Database getInstance() {
        // create object if it's not already created
        if(dbObject == null) {
            dbObject = new Database();
        }

        // returns the singleton object
        return dbObject;
    }

    public void getConnection() {
        System.out.println("You are now connected to the database.");
    }
}
```
- Terminal:** At the bottom, the terminal window shows the command-line interface output:

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
You are now connected to the database.
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```
- Bottom Bar:** The taskbar at the bottom includes icons for file operations, search, and system status.

✓ Java enums

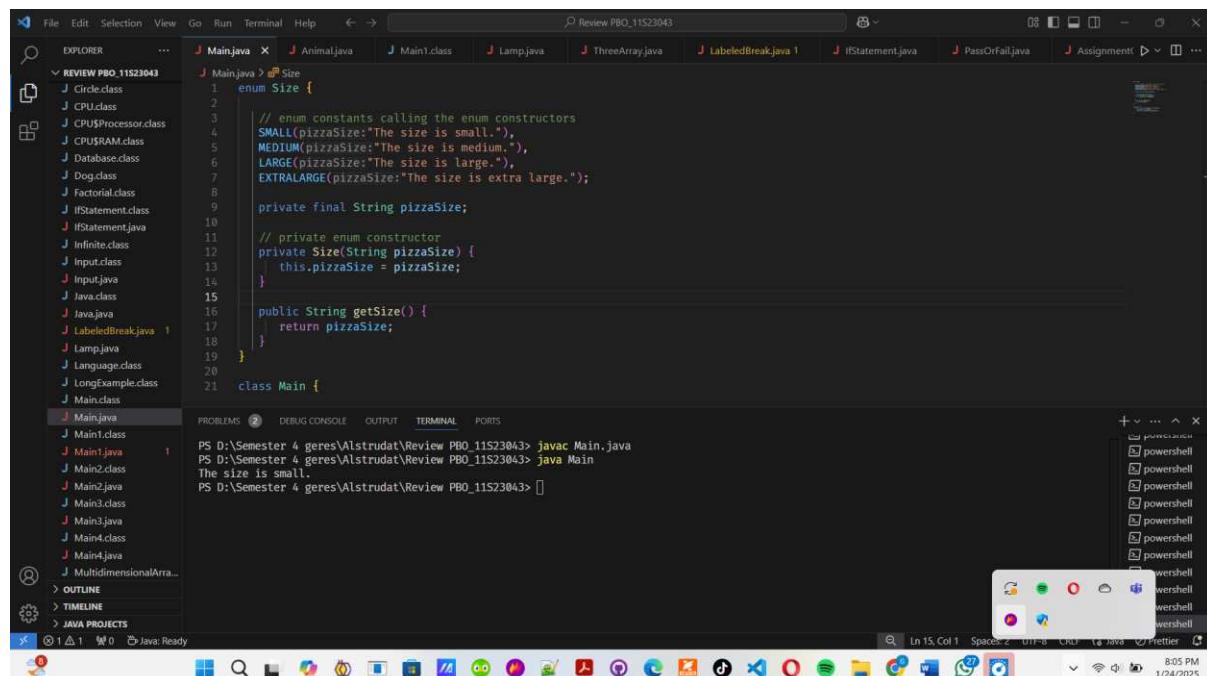
The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows files like Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and AssignmentK.java.
- Code Editor:** The Main.java file is open, containing a main method that prints "SMALL" and "MEDIUM".
- Terminal:** The terminal window shows the command `javac Main.java` followed by the output of the program, which prints "SMALL" and "MEDIUM".
- Status Bar:** Shows "Java: Ready" and other system information.

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java enum Constructor



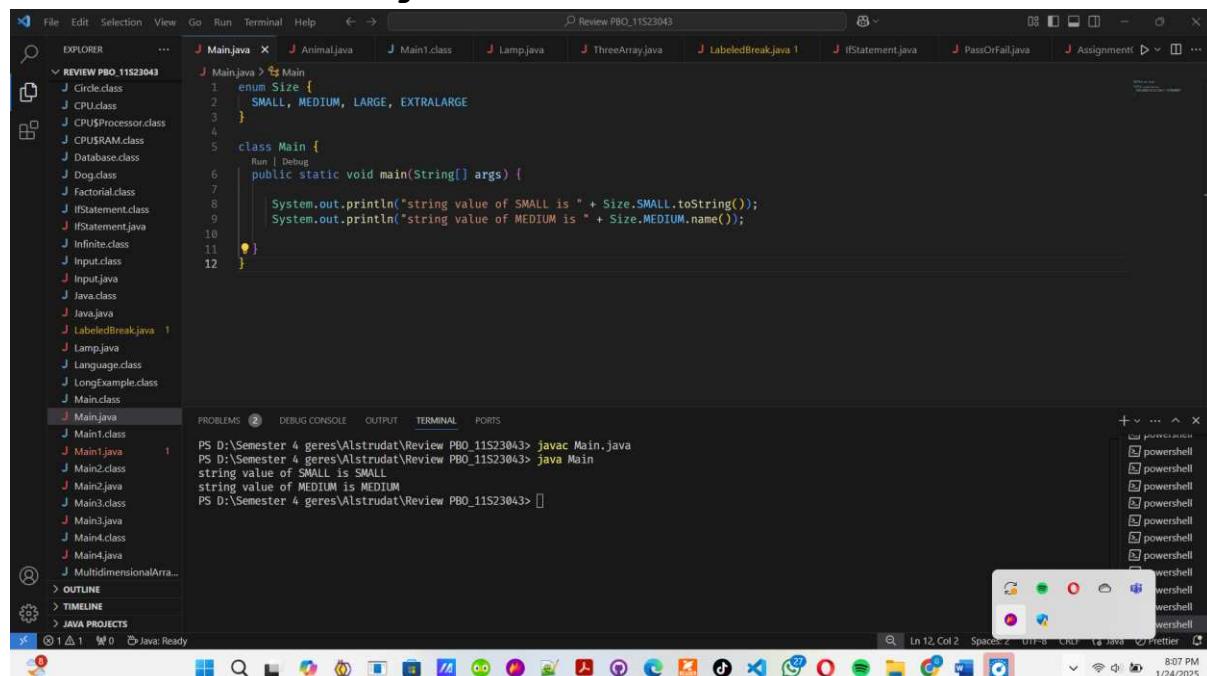
```
File Edit Selection View Go Run Terminal Help ← → Review PBO_11S23043 J Main.java x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D ... EXPLORE J Main.java x J Main.java > Size 1 enum Size { 2 // enum constants calling the enum constructors 3 SMALL(pizzaSize:"The size is small."), 4 MEDIUM(pizzaSize:"The size is medium."), 5 LARGE(pizzaSize:"The size is large."), 6 EXTRALARGE(pizzaSize:"The size is extra large."); 7 8 private final String pizzaSize; 9 10 // private enum constructor 11 private Size(String pizzaSize) { 12     this.pizzaSize = pizzaSize; 13 } 14 15 public String getSize() { 16     return pizzaSize; 17 } 18 } 19 } 20 21 class Main { 22     public static void main(String[] args) { 23         System.out.println("string value of SMALL is " + Size.SMALL.toString()); 24         System.out.println("string value of MEDIUM is " + Size.MEDIUM.name()); 25     } 26 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
The size is small.
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java Ready

✓ Java enum Strings



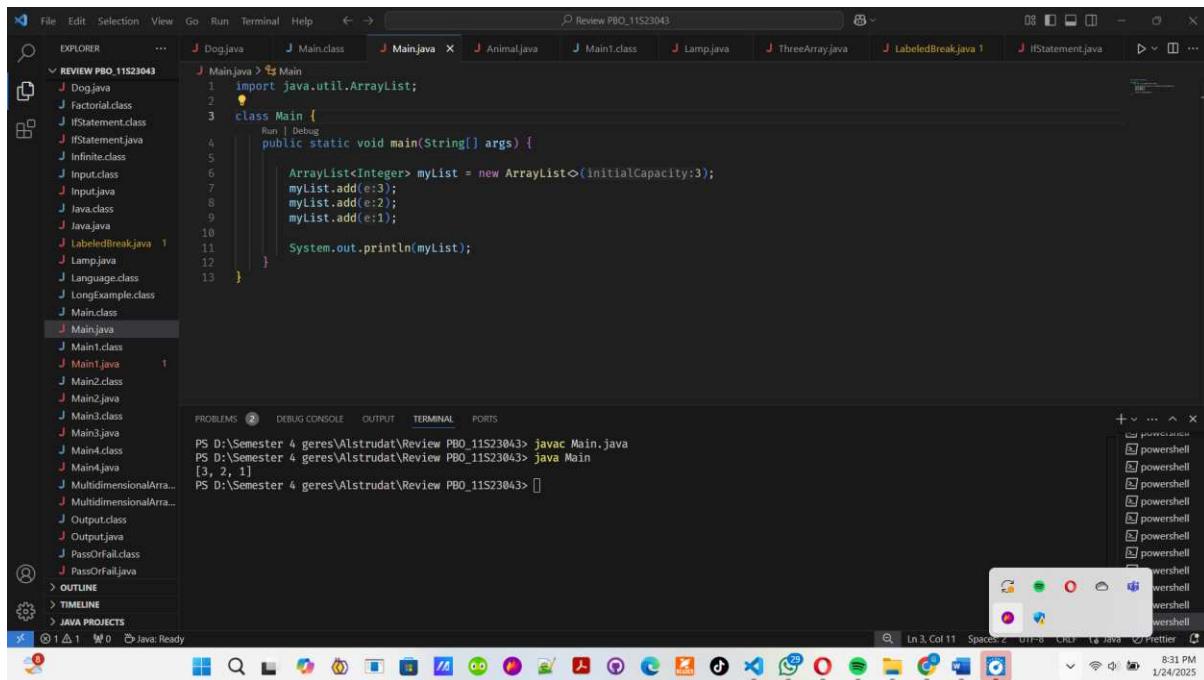
```
File Edit Selection View Go Run Terminal Help ← → Review PBO_11S23043 J Main.java x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D ... EXPLORE J Main.java x J Main.java > Main 1 enum Size { 2     SMALL, MEDIUM, LARGE, EXTRALARGE 3 } 4 5 class Main { 6     public static void main(String[] args) { 7         System.out.println("string value of SMALL is " + Size.SMALL.toString()); 8         System.out.println("string value of MEDIUM is " + Size.MEDIUM.name()); 9     } 10 } 11 12 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
string value of SMALL is SMALL
string value of MEDIUM is MEDIUM
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

Java Ready

✓ Java Package

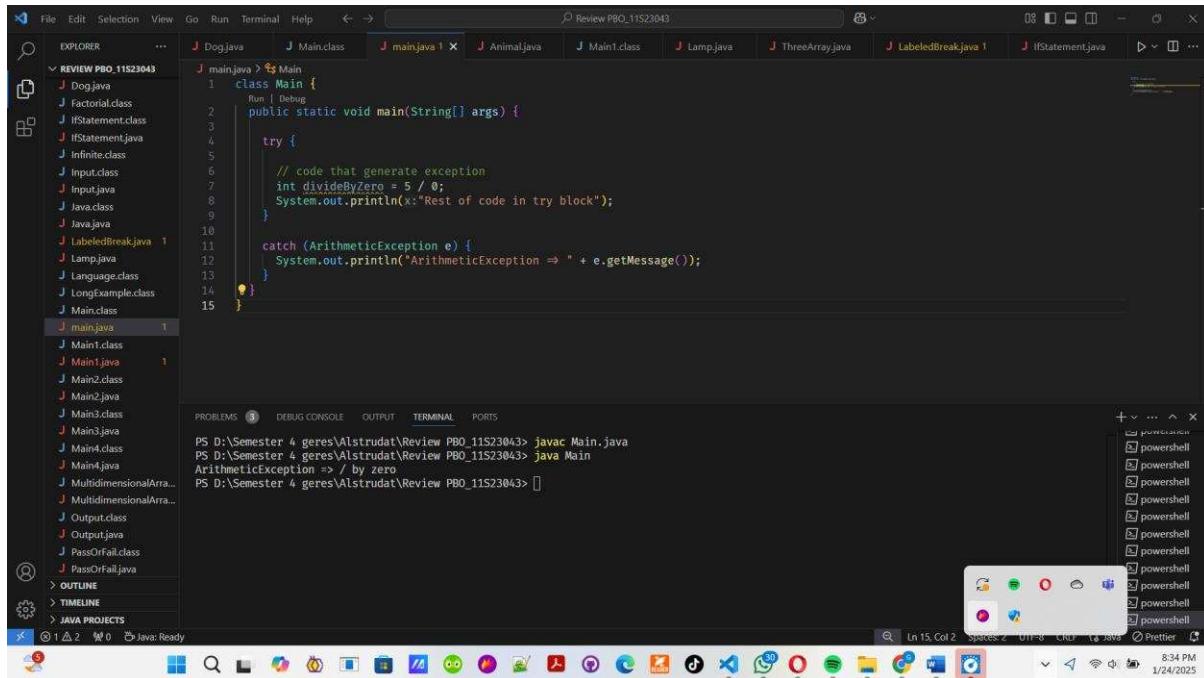


The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The Main.java file is selected and open in the editor. The code prints the elements of an ArrayList. The terminal shows the output: [3, 2, 1].

```
import java.util.ArrayList;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> myList = new ArrayList<>(InitialCapacity:3);
        myList.add(e3);
        myList.add(e2);
        myList.add(e1);
        System.out.println(myList);
    }
}
```

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
[3, 2, 1]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

● Java Exception Handling



The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The main.java file is selected and open in the editor. It contains a try-catch block that divides 5 by 0, generating an ArithmeticException. The terminal shows the output: ArithmeticException => / by zero.

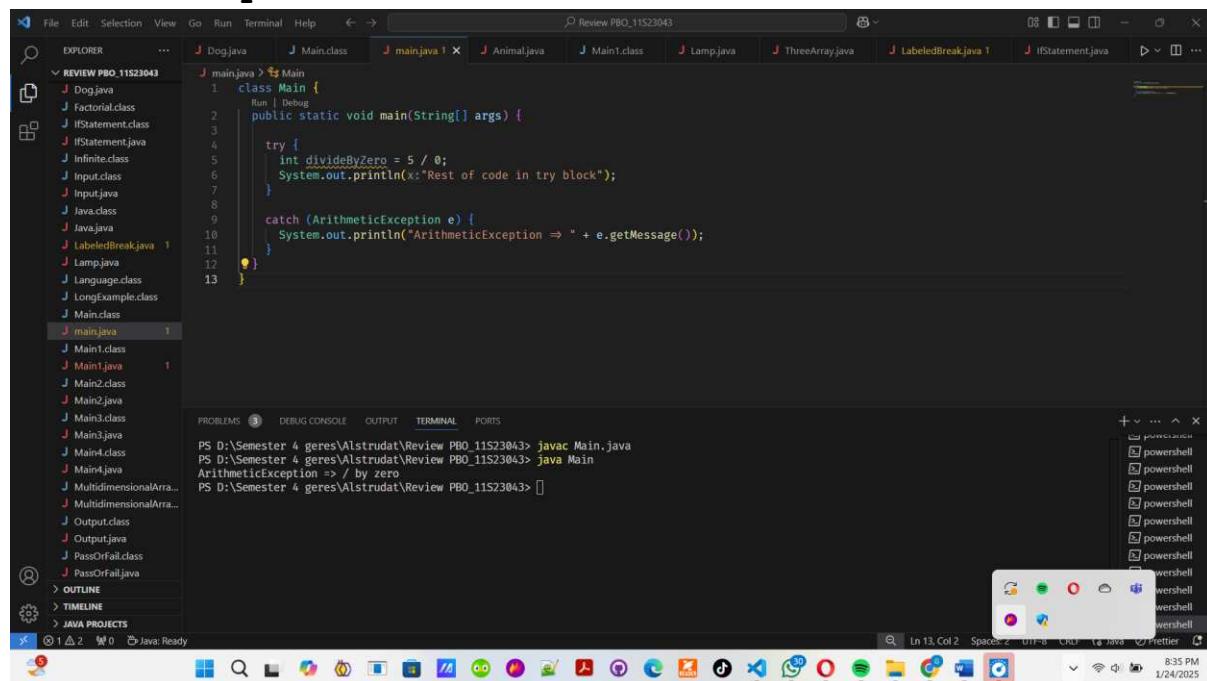
```
class Main {
    public static void main(String[] args) {
        try {
            // code that generate exception
            int divideByZero = 5 / 0;
            System.out.println("Rest of code in try block");
        } catch (ArithmaticException e) {
            System.out.println("ArithmaticException => " + e.getMessage());
        }
    }
}
```

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
ArithmaticException => / by zero
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java try...catch



A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "Review PBO_11S23043". The left sidebar shows a project tree with files like Dog.java, IfStatement.class, and main.java. The main editor window contains the following Java code:

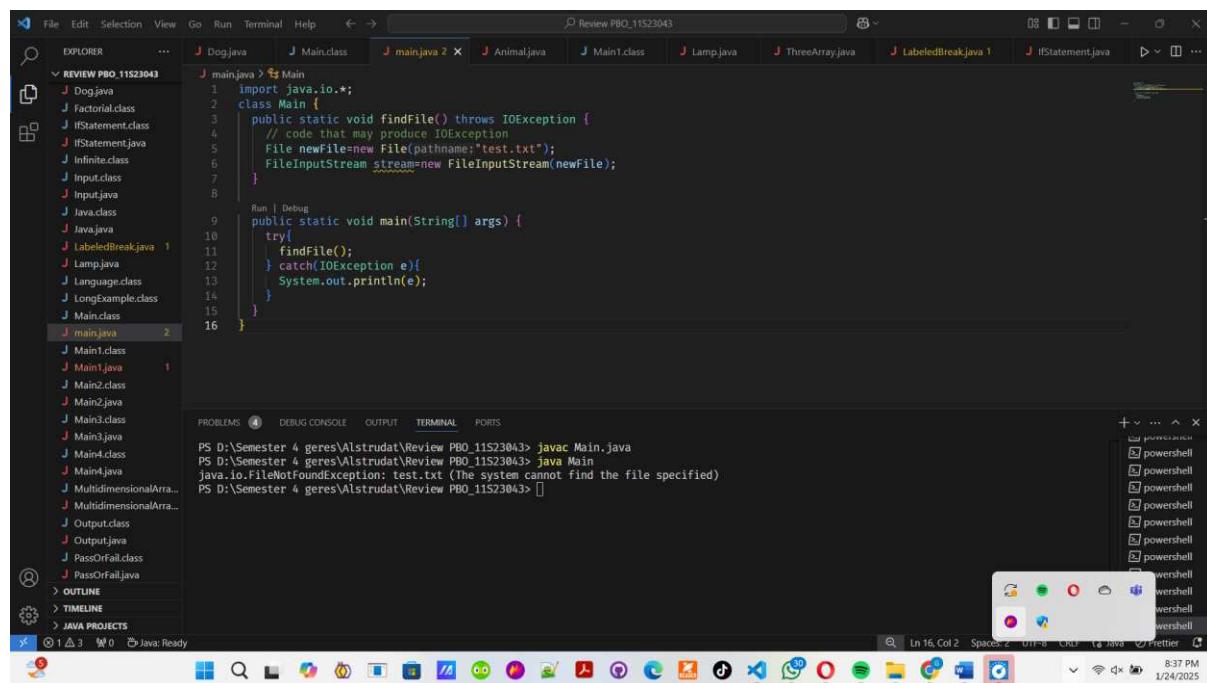
```
REVIEW PBO_11S23043
J Dog.java J Main.class J main.java 1 x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java

J main.java > Main
1 class Main {
2     public static void main(String[] args) {
3         try {
4             int divideByZero = 5 / 0;
5             System.out.println("Rest of code in try block");
6         }
7         catch (ArithmaticException e) {
8             System.out.println("ArithmaticException => " + e.getMessage());
9         }
10    }
11 }
12 }
13 }
```

The terminal at the bottom shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
ArithmaticException => / by zero
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [
```

✓ Java throw and throws



A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "Review PBO_11S23043". The left sidebar shows a project tree with files like Dog.java, IfStatement.class, and main.java. The main editor window contains the following Java code:

```
REVIEW PBO_11S23043
J Dog.java J Main.class J main.java 2 x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java

J main.java > Main
1 import java.io.*;
2 class Main {
3     public static void findFile() throws IOException {
4         // code that may produce IOException
5         File newFile=new File(pathname:"test.txt");
6         FileInputStream stream=new FileInputStream(newFile);
7     }
8
9     public static void main(String[] args) {
10        try{
11            findFile();
12        } catch(IOException e){
13            System.out.println(e);
14        }
15    }
16 }
```

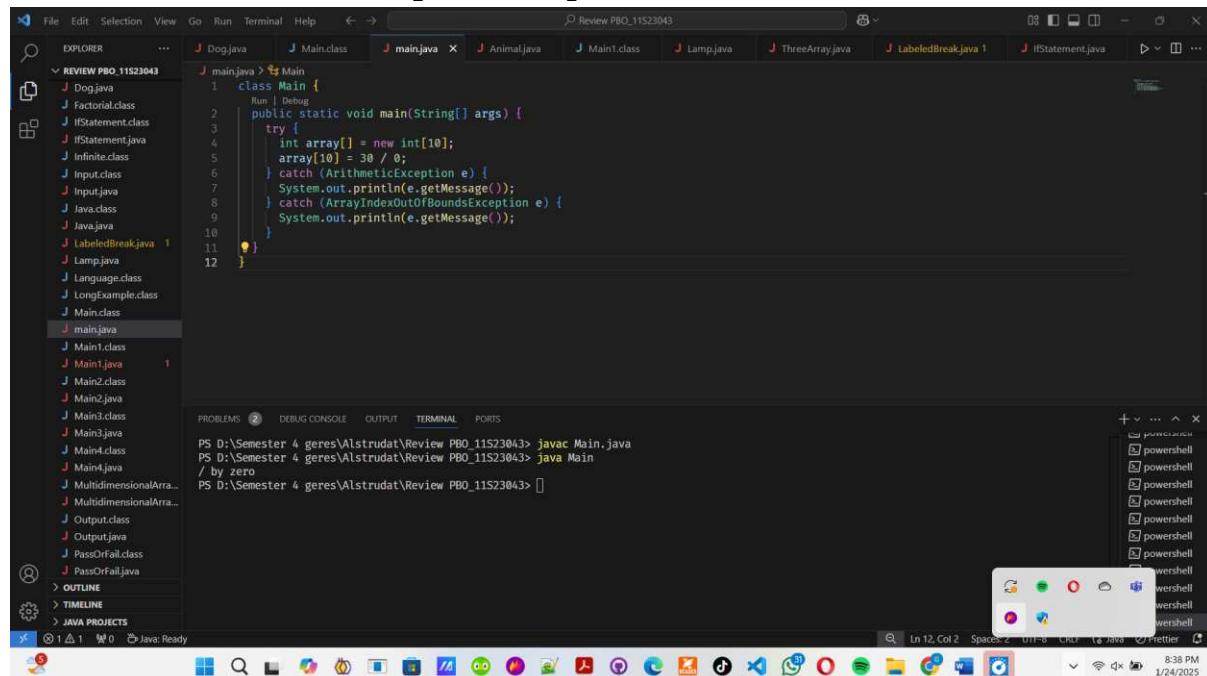
The terminal at the bottom shows the command line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
java.io.FileNotFoundException: test.txt (The system cannot find the file specified)
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java catch Multiple Exceptions



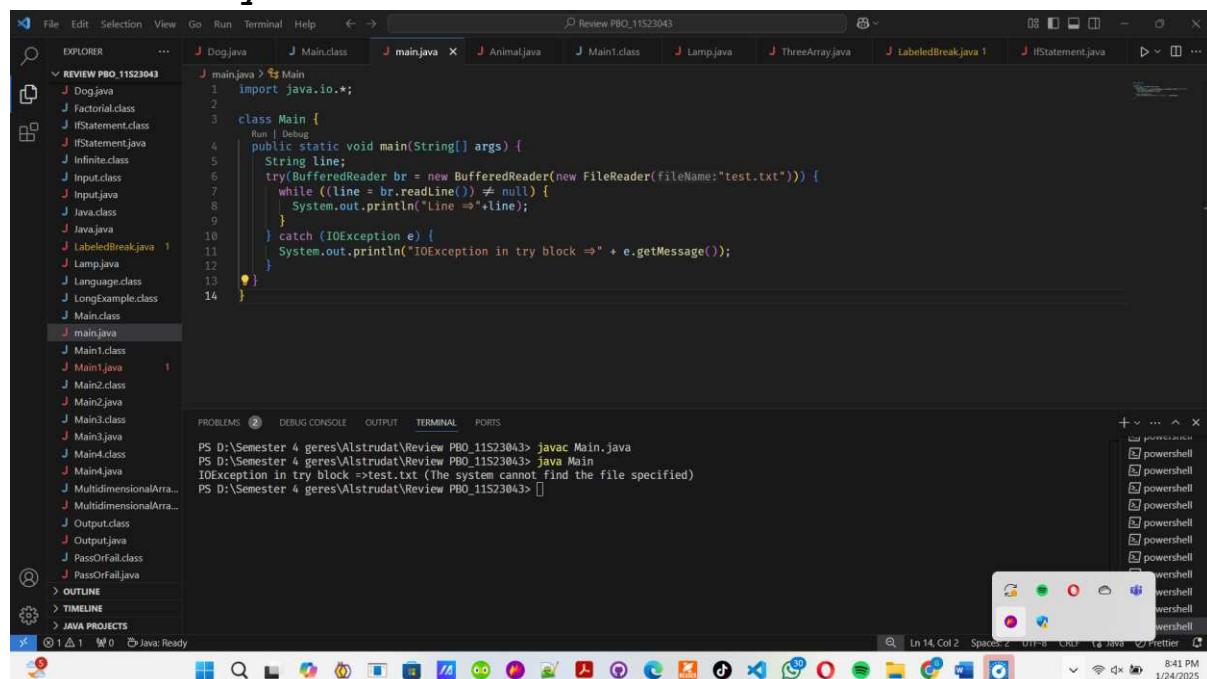
The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The main.java file is open in the editor. The code demonstrates catching multiple exceptions:

```
1 class Main {
2     public static void main(String[] args) {
3         try {
4             int array[] = new int[10];
5             array[10] = 30 / 0;
6         } catch (ArithmetricException e) {
7             System.out.println(e.getMessage());
8         } catch (ArrayIndexOutOfBoundsException e) {
9             System.out.println(e.getMessage());
10        }
11    }
12 }
```

The Problems panel shows no errors. The terminal output shows the execution of the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
/ by zero
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

✓ Java try-with-resources



The screenshot shows a Java project named "REVIEW PBO_11S23043" in the Explorer view. The main.java file is open in the editor. The code uses a try-with-resources block to handle file reading:

```
1 import java.io.*;
2
3 class Main {
4     public static void main(String[] args) {
5         String line;
6         try(BufferedReader br = new BufferedReader(new FileReader(fileName:"test.txt"))){
7             while ((line = br.readLine()) != null) {
8                 System.out.println("Line =>" + line);
9             }
10        } catch (IOException e) {
11            System.out.println("IOException in try block =>" + e.getMessage());
12        }
13    }
14 }
```

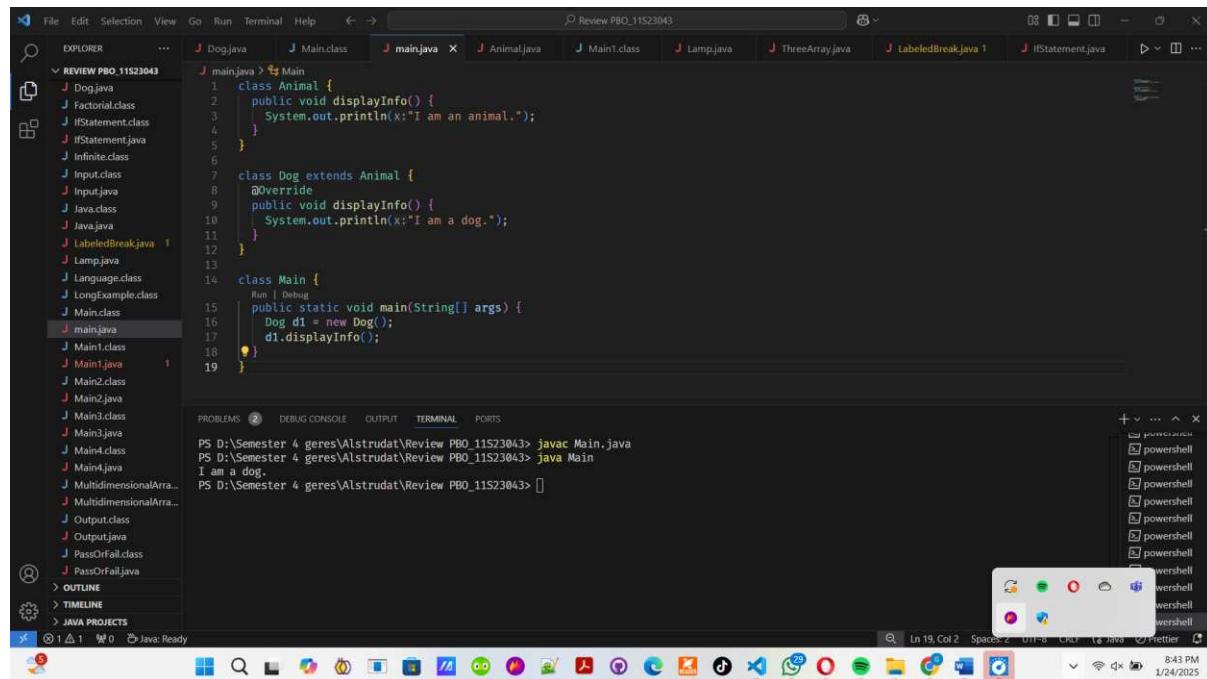
The Problems panel shows no errors. The terminal output shows the execution of the code, which fails because the file does not exist:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
IOException in try block =>test.txt (The system cannot find the file specified)
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java Annotations



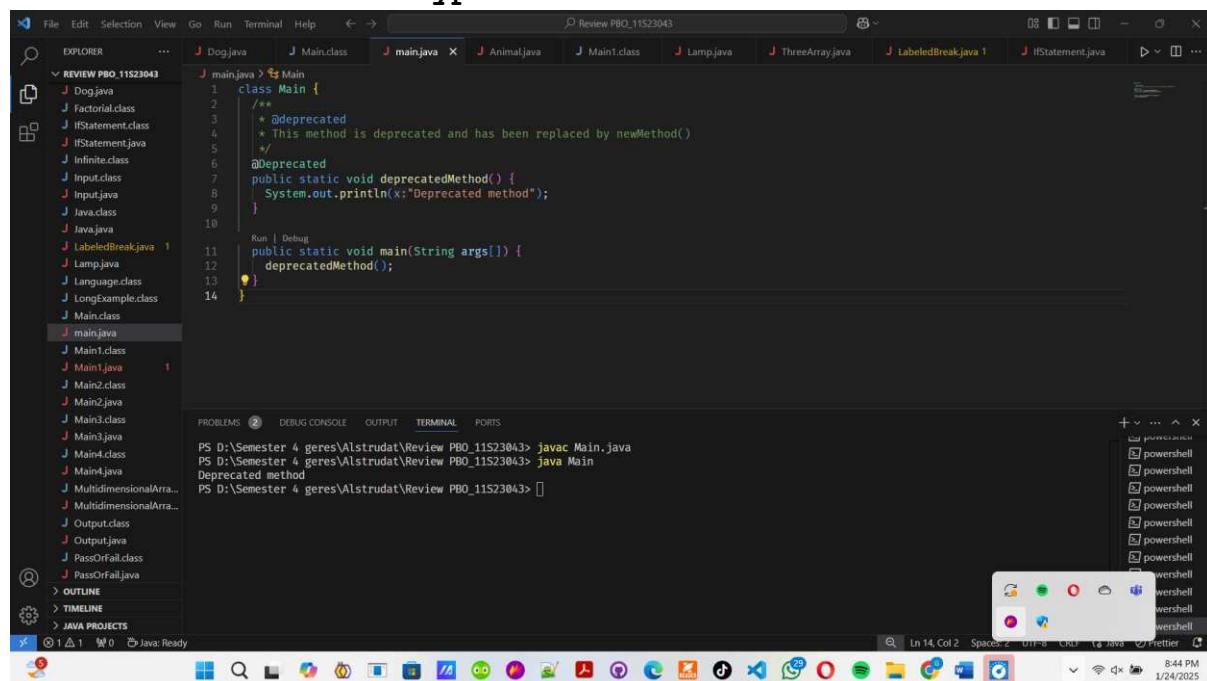
A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project tree with files like Dog.java, Main.class, main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, and others. The main editor window displays Java code:

```
main.java > Main
1 class Animal {
2     public void displayInfo() {
3         System.out.println(x:"I am an animal.");
4     }
5 }
6
7 class Dog extends Animal {
8     @Override
9     public void displayInfo() {
10         System.out.println(x:"I am a dog.");
11     }
12 }
13
14 class Main {
15     Run | Debug
16     public static void main(String[] args) {
17         Dog d1 = new Dog();
18         d1.displayInfo();
19     }
}
```

The terminal at the bottom shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
I am a dog.
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

✓ Java Annotation Types



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project tree with files like Dog.java, Main.class, main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, and others. The main editor window displays Java code:

```
main.java > Main
1 /**
2 * @deprecated
3 * This method is deprecated and has been replaced by newMethod().
4 */
5 @Deprecated
6 public static void deprecatedMethod() {
7     System.out.println(x:"Deprecated method");
8 }
9
10 Run | Debug
11 public static void main(String args[]) {
12     deprecatedMethod();
13 }
14
```

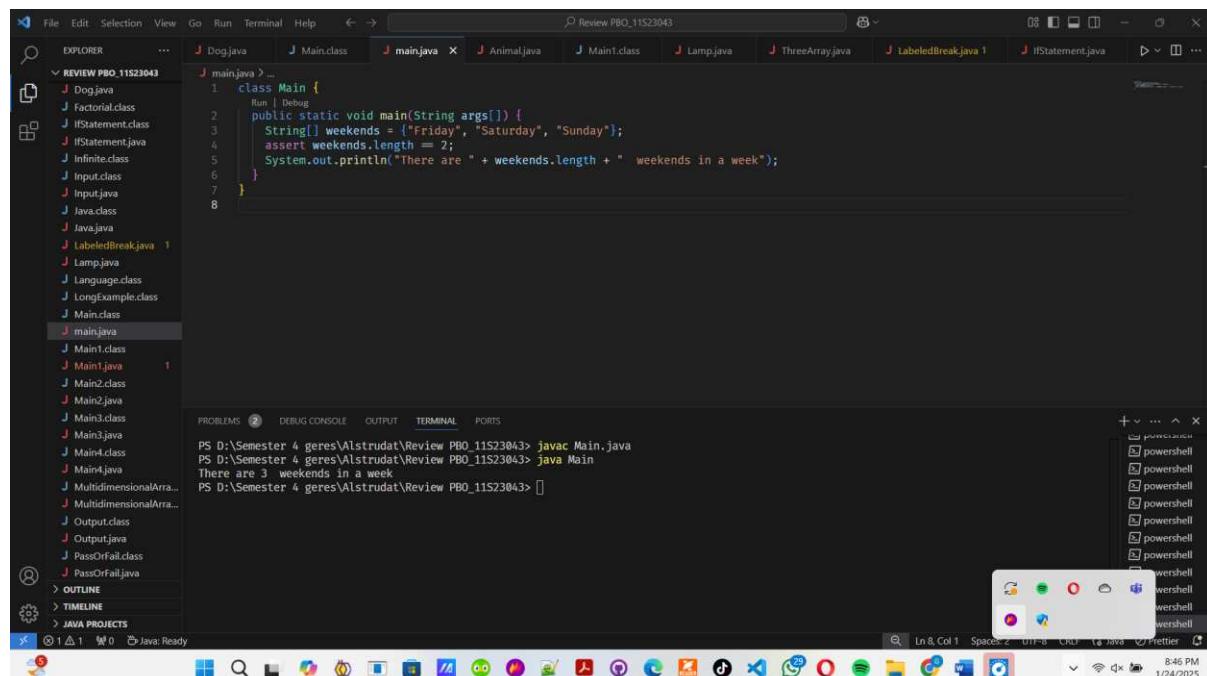
The terminal at the bottom shows the output of running the code:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Deprecated method
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

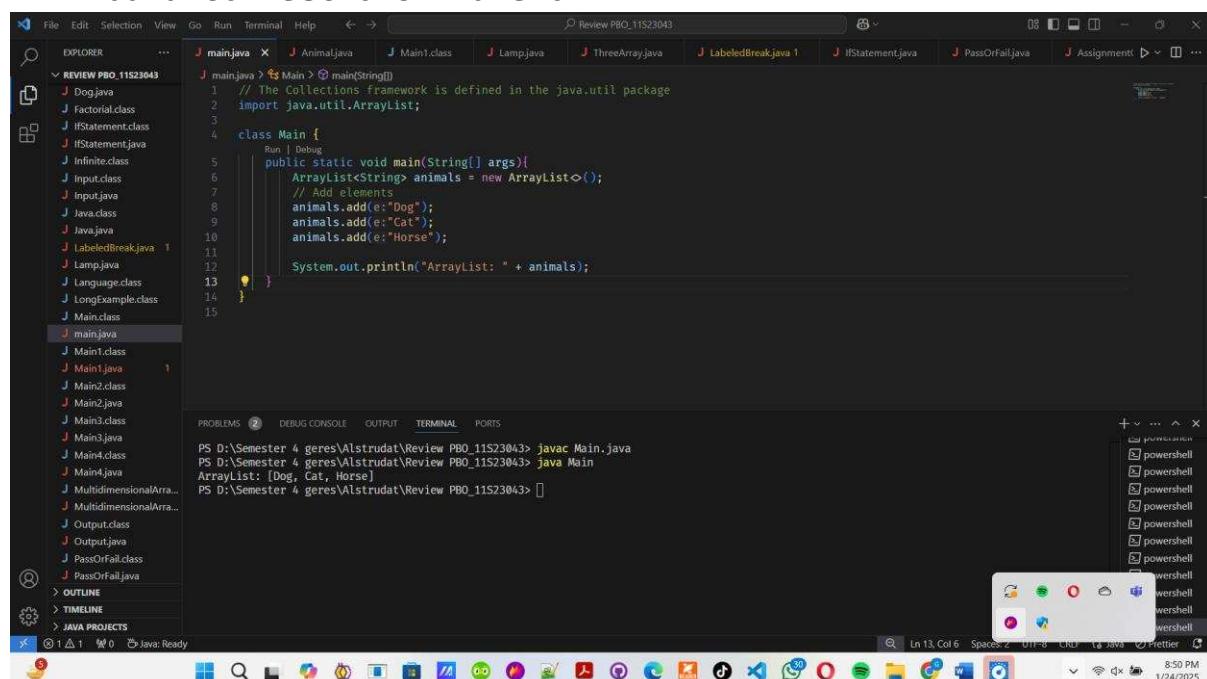
✓ Java Assertions



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043  
EXPLORER J Dog.java J Main.class J main.java X J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J Infinite.class J Input.class J Input.java J Java.class J Java.java J LabeledBreak.java 1 J Lamp.java J Language.class J LongExample.class J Main.class J main.java J Main1.class J Main1.java 1 J Main2.class J Main2.java J Main3.class J Main3.java J Main4.class J Main4.java J MultidimensionalArra... J MultidimensionalArra... J Output.class J Output.java J PassOrFail.class J PassOrFail.java J OUTLINE J TIMELINE J JAVA PROJECTS  
J main.java > ~ 1 class Main { 2     public static void main(String args[]) { 3         String[] weekends = {"Friday", "Saturday", "Sunday"}; 4         assert weekends.length == 2; 5         System.out.println("There are " + weekends.length + " weekends in a week"); 6     } 7 } 8  
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
There are 3 weekends in a week  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]  
Ln 8, Col 1 Spaces: 2 011-5 CWD To Java ⌂ Fetter ⌂ 8:46 PM 1/24/2025
```

✓ Java List

✓ Java Collections Framework



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043  
EXPLORER J Dog.java J Main.class J main.java X J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J Infinite.class J Input.class J Input.java J Java.class J Java.java J LabeledBreak.java 1 J Lamp.java J Language.class J LongExample.class J Main.class J main.java J Main1.class J Main1.java 1 J Main2.class J Main2.java J Main3.class J Main3.java J Main4.class J Main4.java J MultidimensionalArra... J MultidimensionalArra... J Output.class J Output.java J PassOrFail.class J PassOrFail.java J OUTLINE J TIMELINE J JAVA PROJECTS  
J main.java > ~ 1 // The Collections framework is defined in the java.util package 2 import java.util.ArrayList; 3 4 class Main { 5     public static void main(String[] args){ 6         ArrayList<String> animals = new ArrayList<>(); 7         // Add elements 8         animals.add("Dog"); 9         animals.add("Cat"); 10        animals.add("Horse"); 11 12        System.out.println("ArrayList: " + animals); 13    } 14 } 15  
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
ArrayList: [Dog, Cat, Horse]  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> [ ]  
Ln 13, Col 6 Spaces: 2 011-5 CWD To Java ⌂ Fetter ⌂ 8:50 PM 1/24/2025
```

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java List

The screenshot shows a Java code editor interface with the following details:

- File Explorer:** Shows files like EXPLORER, REVIEW PBO_11S2043, and various Java source files (e.g., Dog.java, Main.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, Assignment.java).
- Code Editor:** The main window displays the content of `Main.java`. The code creates a list of integers, adds elements, prints the list, and then removes the second element.
- Terminal:** The terminal window shows the command `javac Main.java` followed by the output of the program's execution, which includes the list [1, 2, 3] and the message "Accessed Element: 2".
- Bottom Bar:** Includes icons for file operations, search, and navigation, along with status information like "Ln 24, Col 2" and "Java: Ready".

✓ Java ArrayList

The screenshot shows a Java code editor interface with the following details:

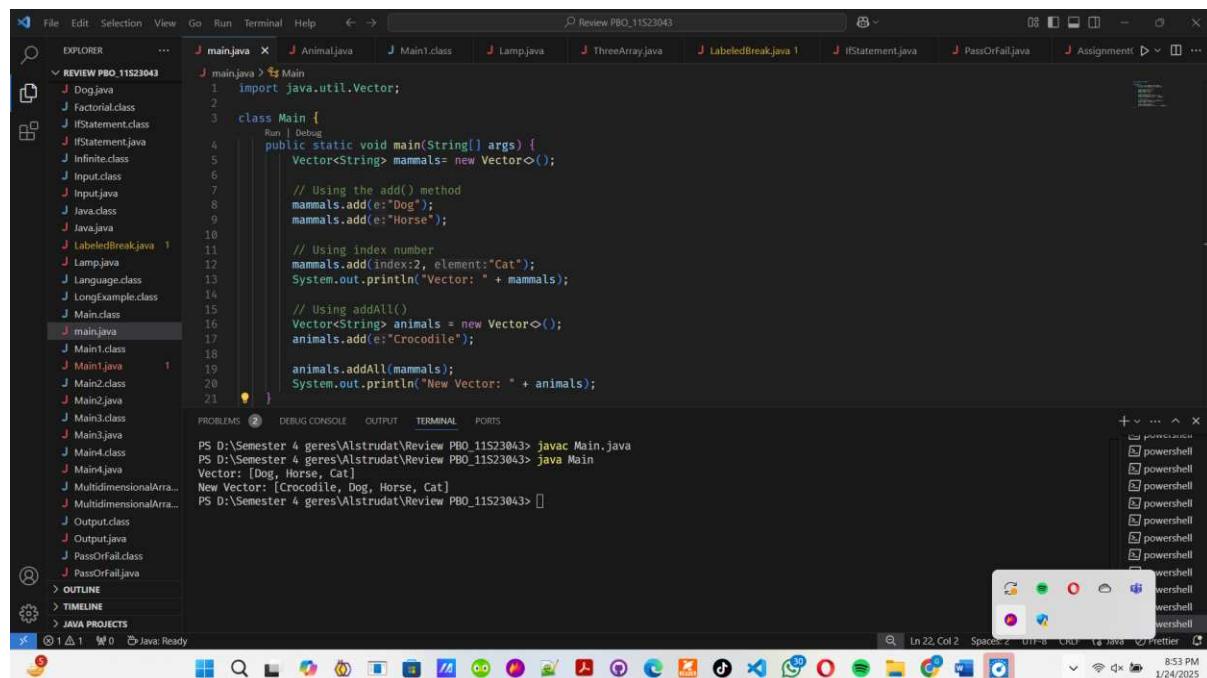
- File Explorer:** Shows a tree view of Java files under "REVIEW PBO_11S23043".
- Code Editor:** The main window displays the content of `Main.java`. The code creates an `ArrayList` named `languages` and adds three elements: "Java", "Python", and "Swift". It then prints the list to the console.
- Terminal:** The bottom terminal window shows the command-line output of running the `Main` class. The output is:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
ArrayList: [Java, Python, Swift]
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> []
```
- Bottom Bar:** Includes icons for file operations, search, and various system status indicators.

11S2215 - Algorithms and Data Structures

Laporan Praktikum

✓ Java Vector



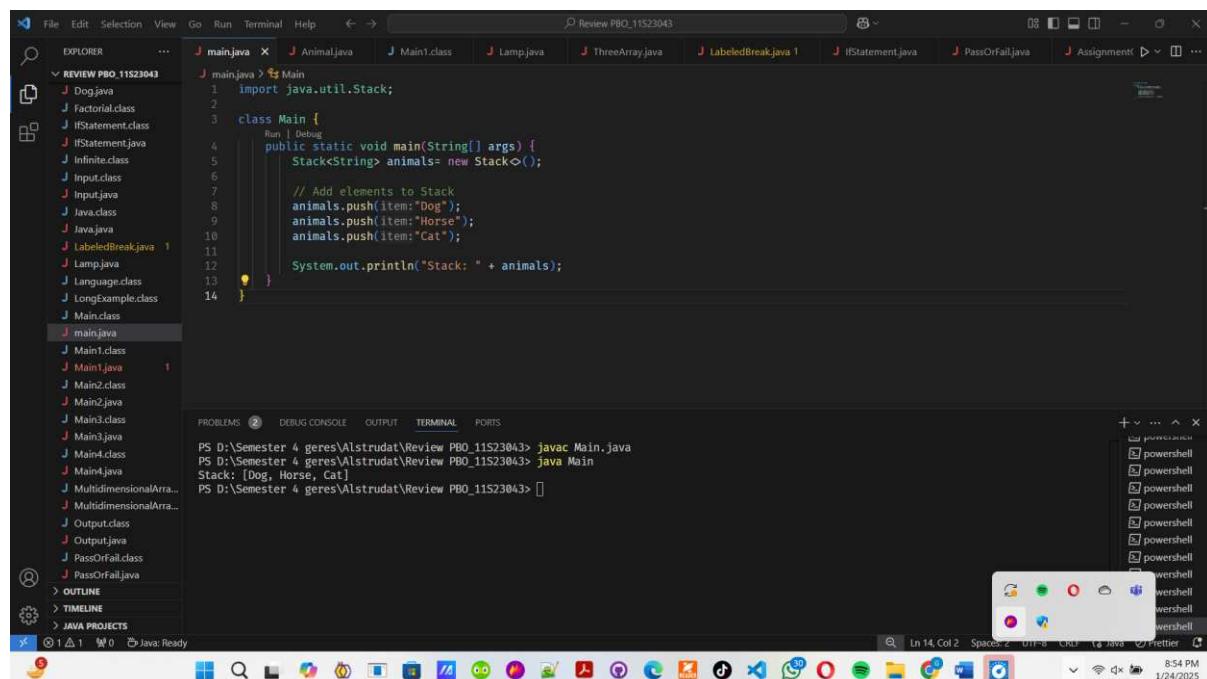
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing various Java files like Dog.java, Factorial.class, IfStatement.class, etc.
- Code Editor:** Displays Java code for the Main class using the Vector class.

```
import java.util.Vector;
class Main {
    public static void main(String[] args) {
        Vector<String> mammals= new Vector<>();
        mammals.add("Dog");
        mammals.add("Horse");
        mammals.add(index:2, element:"Cat");
        System.out.println("Vector: " + mammals);
        Vector<String> animals = new Vector<>();
        animals.add("Crocodile");
        animals.addAll(mammals);
        System.out.println("New Vector: " + animals);
    }
}
```
- Terminal:** Shows the command-line output of running the Main.java file.

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Vector: [Dog, Horse, Cat]
New Vector: [Crocodile, Dog, Horse, Cat]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```
- Bottom Bar:** Shows the Windows taskbar with various pinned icons.

✓ Java Stack Class



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "REVIEW PBO_11S23043" containing various Java files like Dog.java, Factorial.class, IfStatement.class, etc.
- Code Editor:** Displays Java code for the Main class using the Stack class.

```
import java.util.Stack;
class Main {
    public static void main(String[] args) {
        Stack<String> animals= new Stack<>();
        animals.push(item:"Dog");
        animals.push(item:"Horse");
        animals.push(item:"Cat");
        System.out.println("Stack: " + animals);
    }
}
```
- Terminal:** Shows the command-line output of running the Main.java file.

```
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Stack: [Dog, Horse, Cat]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```
- Bottom Bar:** Shows the Windows taskbar with various pinned icons.

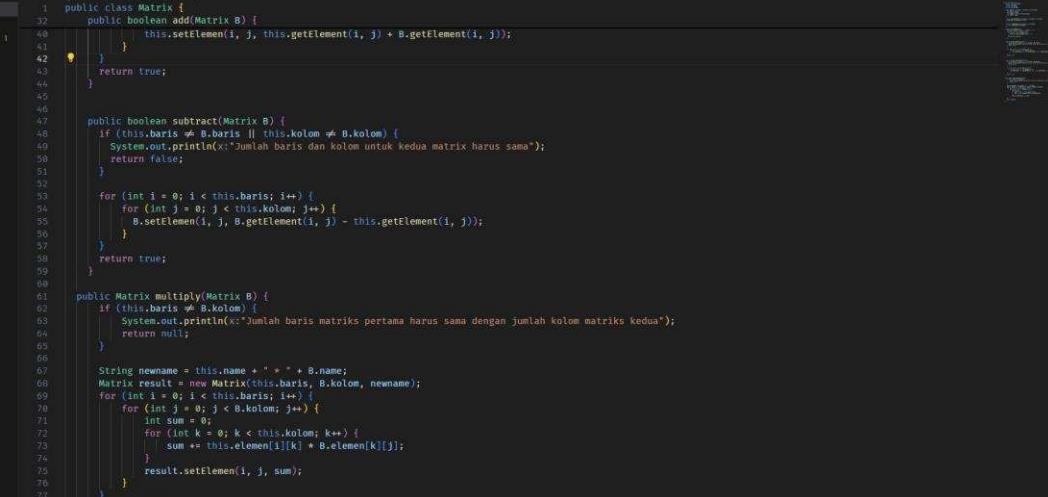
11S2215 - Algorithms and Data Structures

Laporan Praktikum

2. Studi Kasus 1: Matrix App

Buatlah kode program untuk matrix dengan operasi agar dapat menampilkan, menjumlahkan, mengurangkan dan mengalikan matrix. Program yang dibuat harus berada pada dua (2) file java yaitu Matrix.java dan MatrixApp.java, silahkan menggunakan perintah berikut untuk menjalankan kode program:

Matrix.java



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

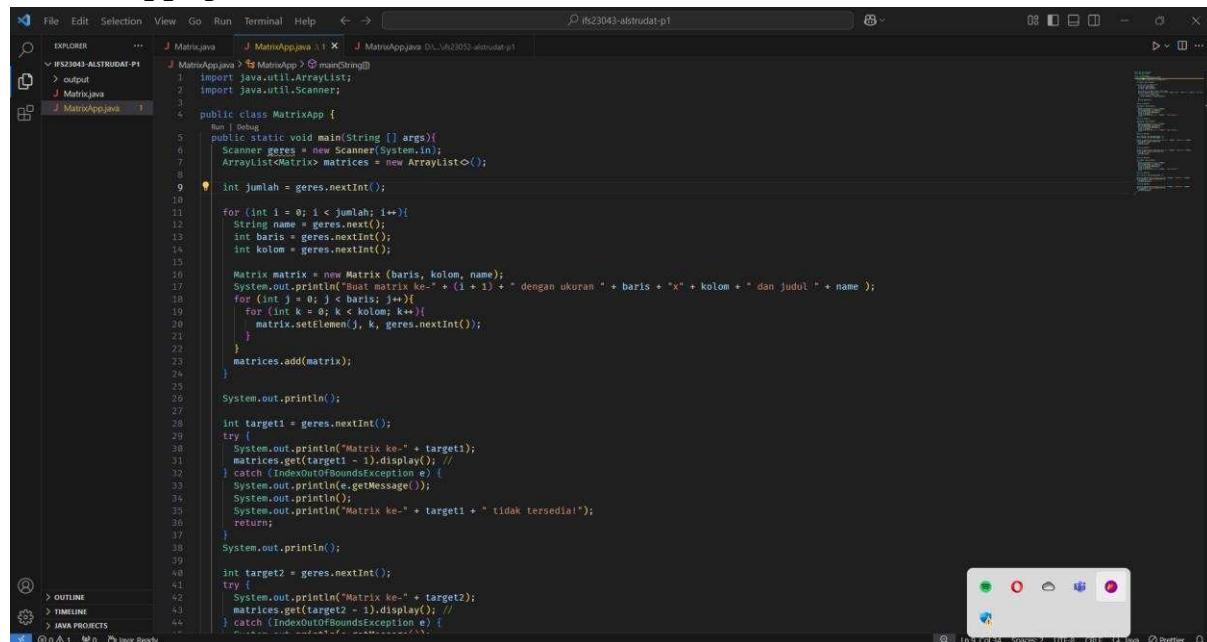
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for Undo, Redo, Cut, Copy, Paste, Find, Select All, and Run.
- Left Sidebar:** Explorer (containing IFS23043_ALSTURDAT_P1, output, Matrix.java, MatrixApp.java), Outline, Timeline, Java Projects.
- Central Area:** Code editor with the Matrix class. The code implements matrix operations like addition, subtraction, and multiplication. It includes validation for matrix dimensions and prints an error message if they don't match.
- Right Sidebar:** Includes tabs for JavaDoc, Properties, and a preview window showing the matrix operations.
- Bottom Status Bar:** Shows the status "Task Ready".

```
File Edit Selection View Go Run Terminal Help < > ihs23043-alsturdat-p1
J Matrix x J MatrixApp.java \ J MatrixApp.java 0, Ihs23043-alsturdat-p1
J Matrix > & Matrix > add(Matrix)
1 public class Matrix {
2     public boolean add(Matrix B) {
3         for (int i = 0; i < this.baris; i++) {
4             for (int j = 0; j < this.kolom; j++) {
5                 this.setElement(i, j, this.getElement(i, j) + B.getElement(i, j));
6             }
7         }
8         return true;
9     }
10
11     public boolean subtract(Matrix B) {
12         if (this.baris != B.baris || this.kolom != B.kolom) {
13             System.out.println("Jumlah baris dan kolom untuk kedua matrix harus sama");
14             return false;
15         }
16
17         for (int i = 0; i < this.baris; i++) {
18             for (int j = 0; j < this.kolom; j++) {
19                 B.setElement(i, j, B.getElement(i, j) - this.getElement(i, j));
20             }
21         }
22         return true;
23     }
24
25     public Matrix multiply(Matrix B) {
26         if (this.baris != B.kolom) {
27             System.out.println("Jumlah baris matriks pertama harus sama dengan jumlah kolom matriks kedua");
28             return null;
29         }
30
31         String newname = this.name + " * " + B.name;
32         Matrix result = new Matrix(this.baris, B.kolom, newname);
33         for (int i = 0; i < this.baris; i++) {
34             for (int j = 0; j < B.kolom; j++) {
35                 int sum = 0;
36                 for (int k = 0; k < this.kolom; k++) {
37                     sum += this.getElement(i, k) * B.getElement(k, j);
38                 }
39                 result.setElement(i, j, sum);
40             }
41         }
42         return result;
43     }
44
45     public void print() {
46         for (int i = 0; i < this.baris; i++) {
47             for (int j = 0; j < this.kolom; j++) {
48                 System.out.print(this.getElement(i, j) + " ");
49             }
50             System.out.println();
51         }
52     }
53
54     public void setElement(int i, int j, double value) {
55         this.element[i][j] = value;
56     }
57
58     public double getElement(int i, int j) {
59         return this.element[i][j];
60     }
61
62     private double[][] element;
63     private int baris, kolom;
64     private String name;
65 }
```

11S2215 - Algorithms and Data Structures

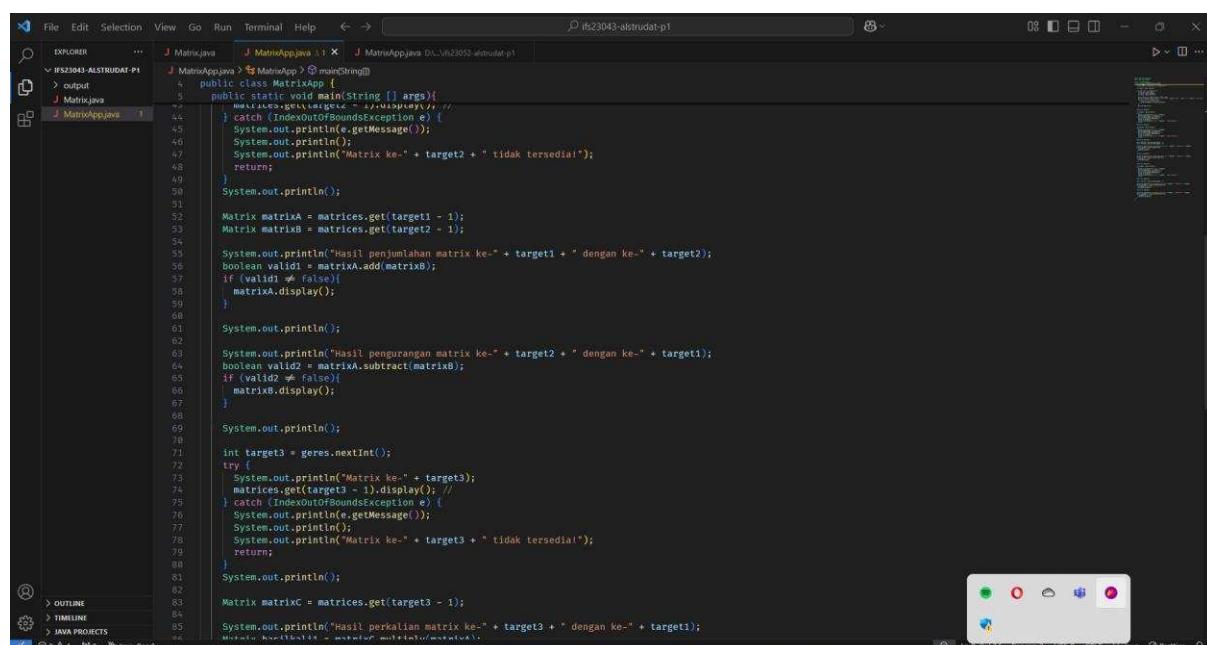
Laporan Praktikum

MatrixApp.java



```
File Edit Selection View Go Run Terminal Help ⏎ → ⌂ ls23043-alstrudat-p1

EXPLORER ... J Matrix.java J MatrixApp.java 1 X J MatrixApp.java D:\Vs23052\alstrudat-p1
VS23043-ALSTRUDAT-P1 J MatrixApp.java > main(String[])
output J Matrix.java
MatrixApp.java 1
public class MatrixApp {
    public static void main(String [] args){
        Scanner geres = new Scanner(System.in);
        ArrayList<Matrix> matrices = new ArrayList<>();
        int jumlah = geres.nextInt();
        for (int i = 0; i < jumlah; i++){
            String name = geres.next();
            int baris = geres.nextInt();
            int kolom = geres.nextInt();
            Matrix matrix = new Matrix(baris, kolom, name);
            System.out.println("Buat matrix ke-" + (i+1) + " dengan ukuran " + baris + "x" + kolom + " dan judul " + name );
            for (int j = 0; j < baris; j++){
                for (int k = 0; k < kolom; k++){
                    matrix.setElement(j, k, geres.nextInt());
                }
            }
            matrices.add(matrix);
        }
        System.out.println();
        int target1 = geres.nextInt();
        try {
            System.out.println("Matrix ke-" + target1);
            matrices.get(target1 - 1).display(); // 
        } catch (IndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
            System.out.println();
            System.out.println("Matrix ke-" + target1 + " tidak tersedia!");
            return;
        }
        System.out.println();
        int target2 = geres.nextInt();
        try {
            System.out.println("Matrix ke-" + target2);
            matrices.get(target2 - 1).display(); // 
        } catch (IndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
            System.out.println();
            System.out.println("Matrix ke-" + target2 + " tidak tersedia!");
            return;
        }
        System.out.println();
        Matrix matrixA = matrices.get(target1 - 1);
        Matrix matrixB = matrices.get(target2 - 1);
        System.out.println("Hasil penjumlahan matrix ke-" + target1 + " dengan ke-" + target2);
        boolean valid1 = matrixA.add(matrixB);
        if (valid1 != false){
            matrixA.display();
        }
        System.out.println();
        System.out.println("Hasil pengurangan matrix ke-" + target2 + " dengan ke-" + target1);
        boolean valid2 = matrixA.subtract(matrixB);
        if (valid2 != false){
            matrixB.display();
        }
        System.out.println();
        int target3 = geres.nextInt();
        try {
            System.out.println("Matrix ke-" + target3);
            matrices.get(target3 - 1).display(); // 
        } catch (IndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
            System.out.println();
            System.out.println("Matrix ke-" + target3 + " tidak tersedia!");
            return;
        }
        System.out.println();
        Matrix matrixC = matrices.get(target3 - 1);
        System.out.println("Hasil perkalian matrix ke-" + target3 + " dengan ke-" + target1);
        Matrix result = matrixA.multiply(matrixB);
        result.display();
    }
}
```



```
File Edit Selection View Go Run Terminal Help ⏎ → ⌂ ls23043-alstrudat-p1

EXPLORER ... J Matrix.java J MatrixApp.java 1 X J MatrixApp.java D:\Vs23052\alstrudat-p1
VS23043-ALSTRUDAT-P1 J MatrixApp.java > main(String[])
output J Matrix.java
MatrixApp.java 1
public class MatrixApp {
    public static void main(String [] args){
        Matrix matrixA = null;
        Matrix matrixB = null;
        Matrix matrixC = null;
        int target1 = geres.nextInt();
        try {
            System.out.println("Matrix ke-" + target1);
            matrixA = matrices.get(target1 - 1);
        } catch (IndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
            System.out.println();
            System.out.println("Matrix ke-" + target1 + " tidak tersedia!");
            return;
        }
        System.out.println();
        System.out.println("Hasil penjumlahan matrix ke-" + target1 + " dengan ke-" + target2);
        boolean valid1 = matrixA.add(matrixB);
        if (valid1 != false){
            matrixA.display();
        }
        System.out.println();
        System.out.println("Hasil pengurangan matrix ke-" + target2 + " dengan ke-" + target1);
        boolean valid2 = matrixA.subtract(matrixB);
        if (valid2 != false){
            matrixB.display();
        }
        System.out.println();
        int target3 = geres.nextInt();
        try {
            System.out.println("Matrix ke-" + target3);
            matrixC = matrices.get(target3 - 1);
        } catch (IndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
            System.out.println();
            System.out.println("Matrix ke-" + target3 + " tidak tersedia!");
            return;
        }
        System.out.println();
        System.out.println("Hasil perkalian matrix ke-" + target3 + " dengan ke-" + target1);
        Matrix result = matrixA.multiply(matrixB);
        result.display();
    }
}
```

Contoh program saat dijalankan:

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a Java project named "IF523043-ALSTRUDAT-P1" containing files: Matrix.java, MatrixApp.java, Matrix.java, and MatrixApp.java.
- Terminal:** Displays the command-line output of running the Java application.

```
PS D:\Semester 4\geres\Alstrudat\if523043-alstrudat-p1> javac -d output MatrixApp.java Matrix.java
PS D:\Semester 4\geres\Alstrudat\if523043-alstrudat-p1> java -cp output MatrixApp

1
2
3
4

1
Matrix ke-1
Matrix A:
1 2
3 4

1
Matrix ke-1
Matrix A:
1 2
3 4

Hasil penjumlahan matrix ke-1 dengan ke-1
Matrix A:
2 4
6 8

Hasil pengurangan matrix ke-1 dengan ke-1
Matrix A:
0 0
0 0

3
Matrix ke-3
Index 2 out of bounds for length 1

Matrix ke-3 tidak tersedia
PS D:\Semester 4\geres\Alstrudat\if523043-alstrudat-p1>
```
- Side Bar:** Shows sections for OUTLINE, TIMELINE, and JAVA PROJECTS.
- Bottom Bar:** Shows the taskbar with various icons.