

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

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

**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

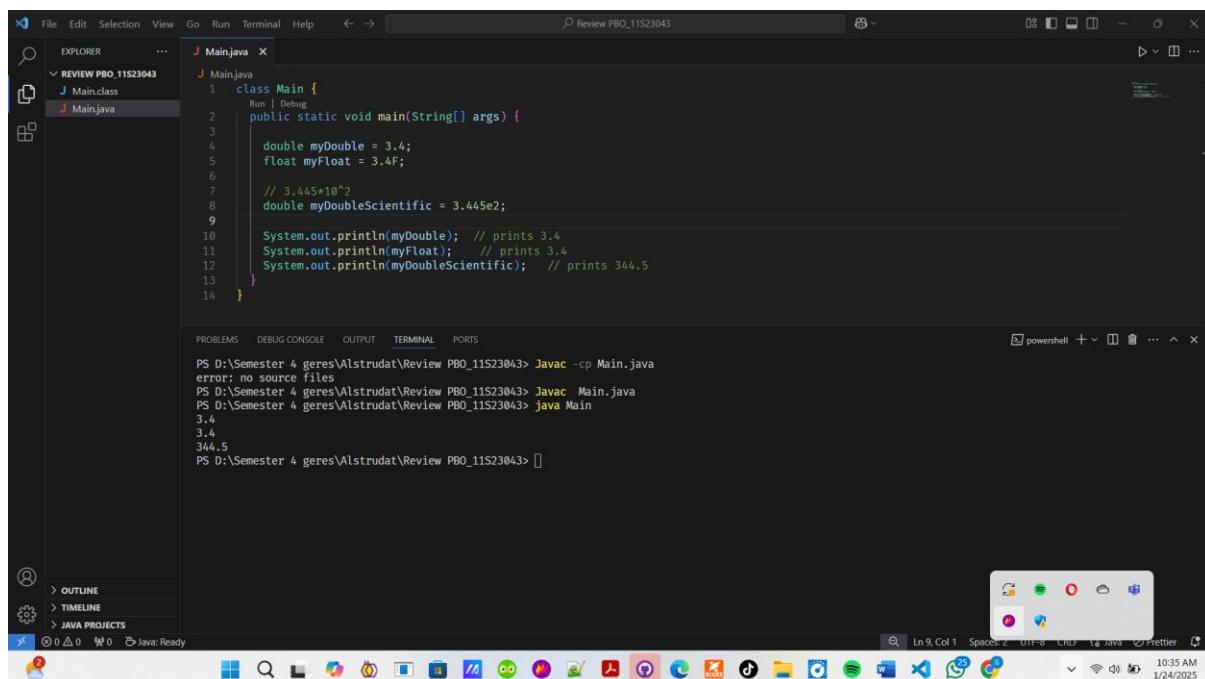
Dalam pemahaman saya, setelah mempelajari dan mempraktikkan praktikum ini, mahasiswa diharapkan mampu memahami kembali konsep pemrograman berorientasi objek (PBO) secara mendalam, khususnya dalam konteks pemrograman Java. Mahasiswa juga diharapkan dapat menerapkan prinsip-prinsip PBO seperti enkapsulasi, 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 code in a file named Main.java:

```
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 }
```

The terminal window below shows the execution of the 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> 
```

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  **$3.445 \times 10^2$**  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 left sidebar 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 of running the code:

```
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 left sidebar 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;  
        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 of running the code:

```
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 Windows desktop environment with the Visual Studio Code (VS Code) application open. The title bar reads "Review PBO\_11S23043". The Explorer sidebar on the left shows files: Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, and Main3.java (selected). The main editor area displays the following Java code:

```
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 tab at the bottom shows the command-line output of running the code:

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

The taskbar at the bottom includes icons for various applications like File Explorer, Task View, and the Start button.

Java int data type

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The title bar reads "Review PBO\_11S23043". The Explorer sidebar on the left shows files: Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, and Main4.java (selected). The main editor area displays the following Java code:

```
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 tab at the bottom shows the command-line output of running the code:

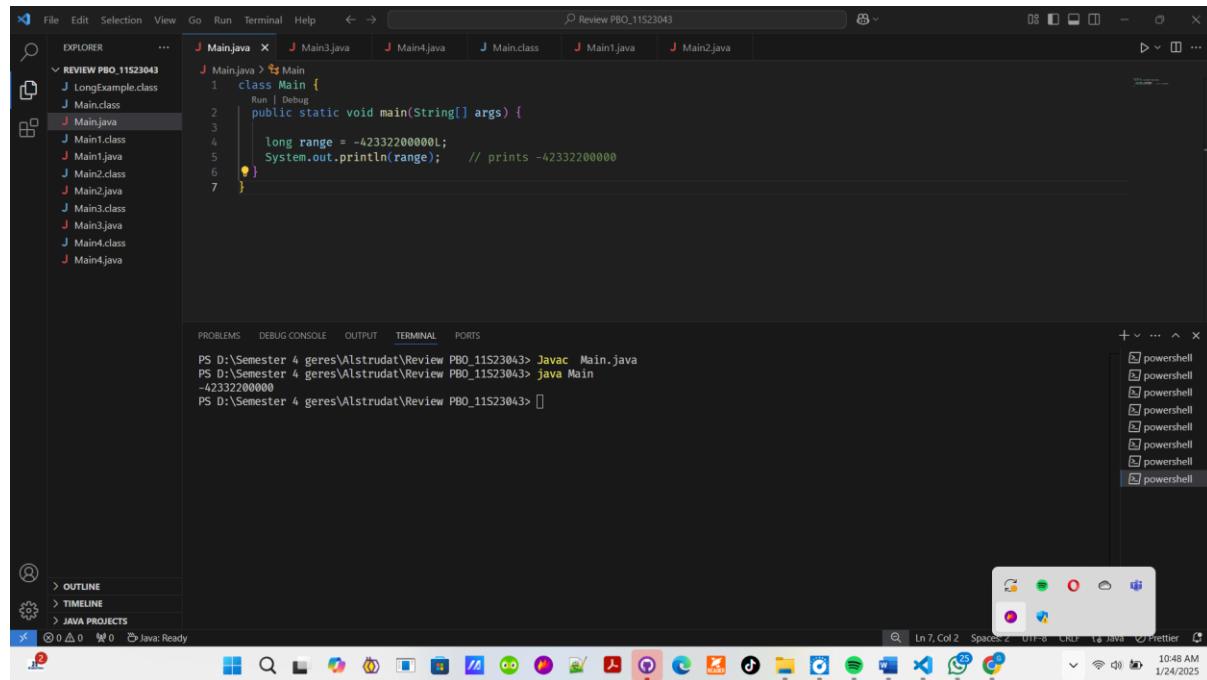
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main4.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main4  
-4250000  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

The taskbar at the bottom includes icons for various applications like File Explorer, Task View, and the Start button.

Java long data type

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

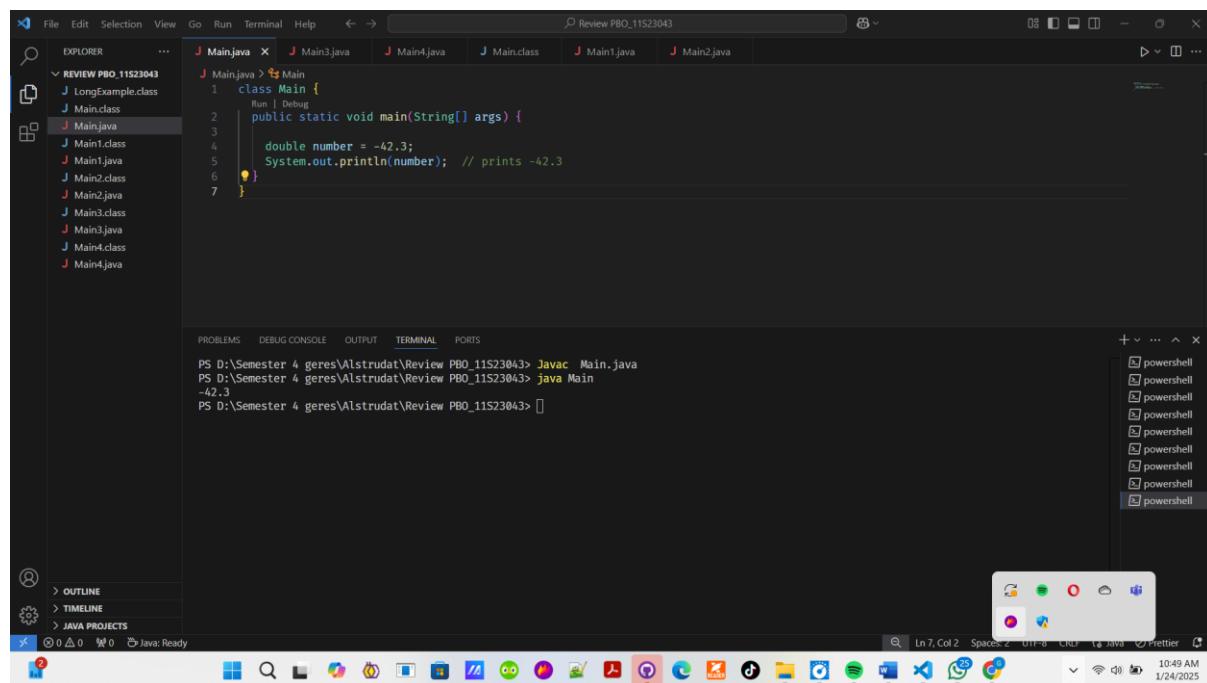


The screenshot shows a Java application named "Main.java" in a code editor. The code prints the value of a long variable "range".

```
class Main {
    public static void main(String[] args) {
        long range = -42332200000L;
        System.out.println(range); // prints -42332200000
    }
}
```

The terminal output shows the command "javac Main.java" followed by "java Main" and the output "-42332200000".

Java double data type



The screenshot shows a Java application named "Main.java" in a code editor. The code prints the value of a double variable "number".

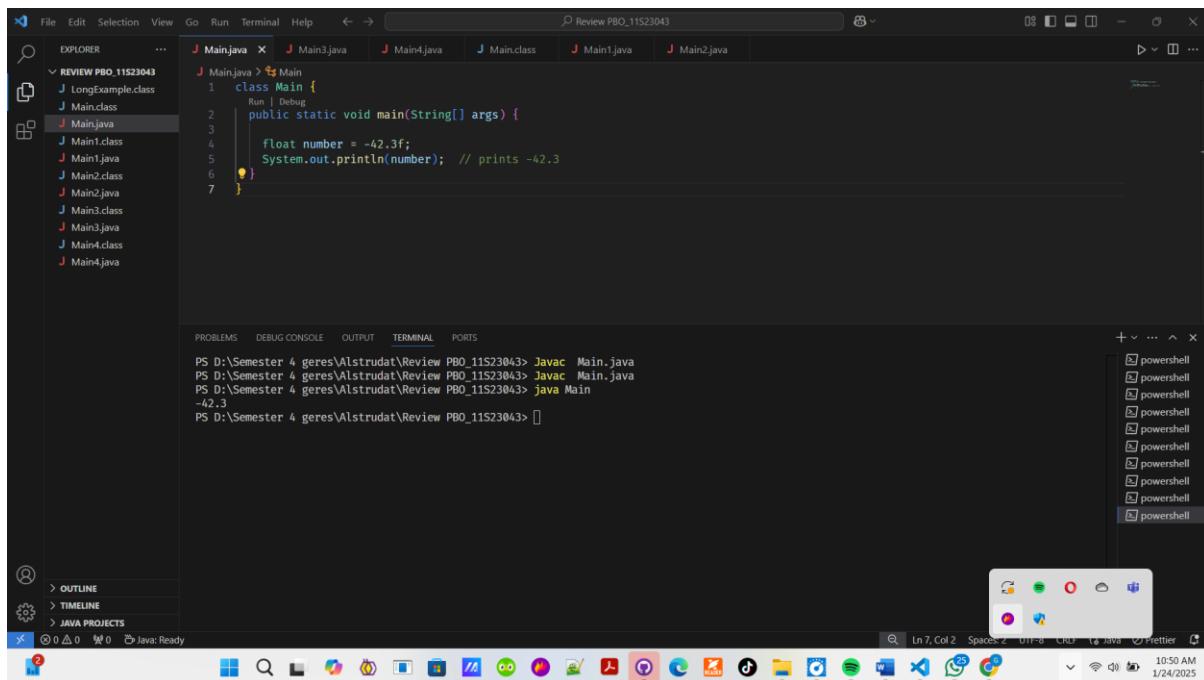
```
class Main {
    public static void main(String[] args) {
        double number = -42.3;
        System.out.println(number); // prints -42.3
    }
}
```

The terminal output shows the command "javac Main.java" followed by "java Main" and the output "-42.3".

Java float data type

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



A screenshot of a Java development environment, likely IntelliJ IDEA or a similar IDE, running on a Windows operating system. The interface shows an Explorer sidebar with files like Main.java, Main1.java, Main2.java, Main3.java, Main4.java, Main1.class, Main2.class, Main3.class, and Main4.class. The main editor window displays the following Java code:

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

The terminal tab at the bottom shows the command-line output of the Java compiler (javac) and the Java runtime (java) executing the Main class:

```
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>
```

The taskbar at the bottom of the screen shows various pinned icons, including Microsoft Edge, File Explorer, and several PowerShell windows.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Java char data type

The screenshot shows a Java project named "REVIEW PBO\_11S23043" in the Explorer sidebar. The "Main.java" file is open in the editor, containing the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         char letter = '\u0051';  
4         System.out.println(letter); // prints Q  
5     }  
6 }
```

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  
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 sidebar. The "Main.java" file is open in the editor, containing the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         // declare variables  
4         int a = 12, b = 5;  
5  
6         // addition operator  
7         System.out.println("a + b = " + (a + b));  
8  
9         // subtraction operator  
10        System.out.println("a - b = " + (a - b));  
11  
12         // multiplication operator  
13        System.out.println("a * b = " + (a * b));  
14  
15         // division operator  
16        System.out.println("a / b = " + (a / b));  
17  
18         // modulo operator  
19        System.out.println("a % b = " + (a % b));  
20  
21    }  
22 }
```

The terminal below shows the output of running the code:

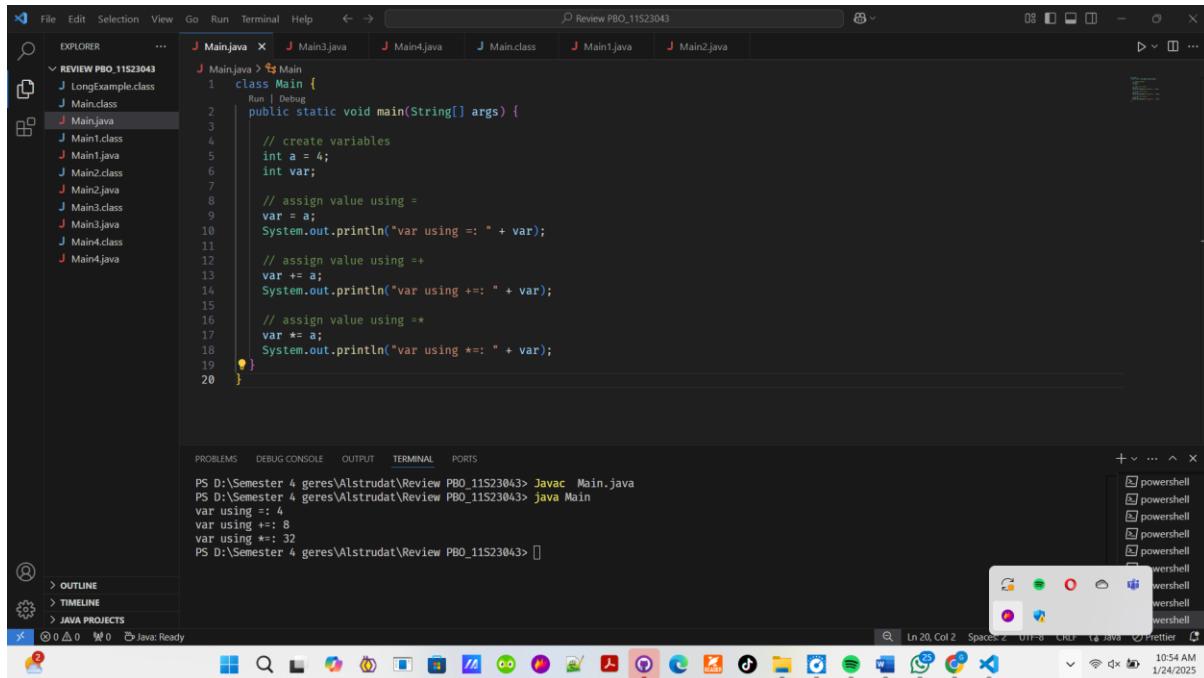
```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
a + b = 17  
a - b = 7  
a * b = 60  
a / b = 2  
a % b = 2  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Assignment Operators



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

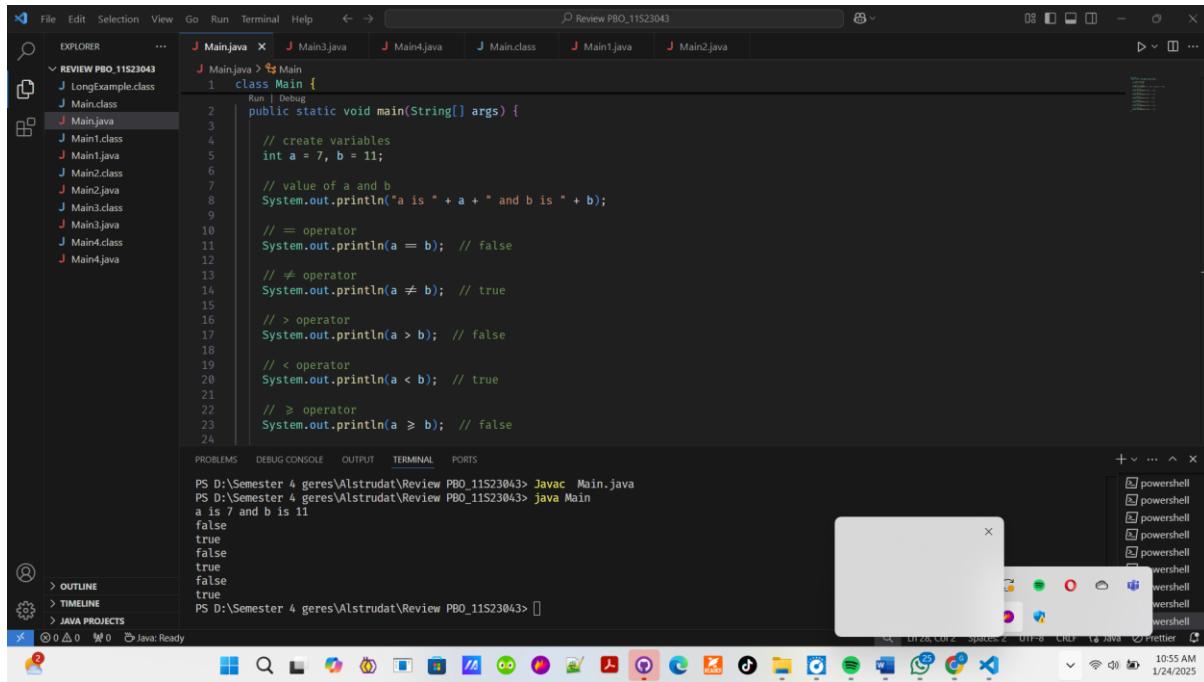
```
1 class Main {  
2     public static void main(String[] args) {  
3         // create variables  
4         int a = 4;  
5         int var;  
6  
7         // assign value using =  
8         var = a;  
9         System.out.println("var using := " + var);  
10  
11        // assign value using +=  
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 below the code editor 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  
var using := 4  
var using +=: 8  
var using *=: 32
```

The status bar at the bottom right indicates the date and time as 10:54 AM on 1/24/2025.

### Relational 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:

```
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 below the code editor 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  
a is 7 and b is 11  
false  
true  
false  
true  
false  
true  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

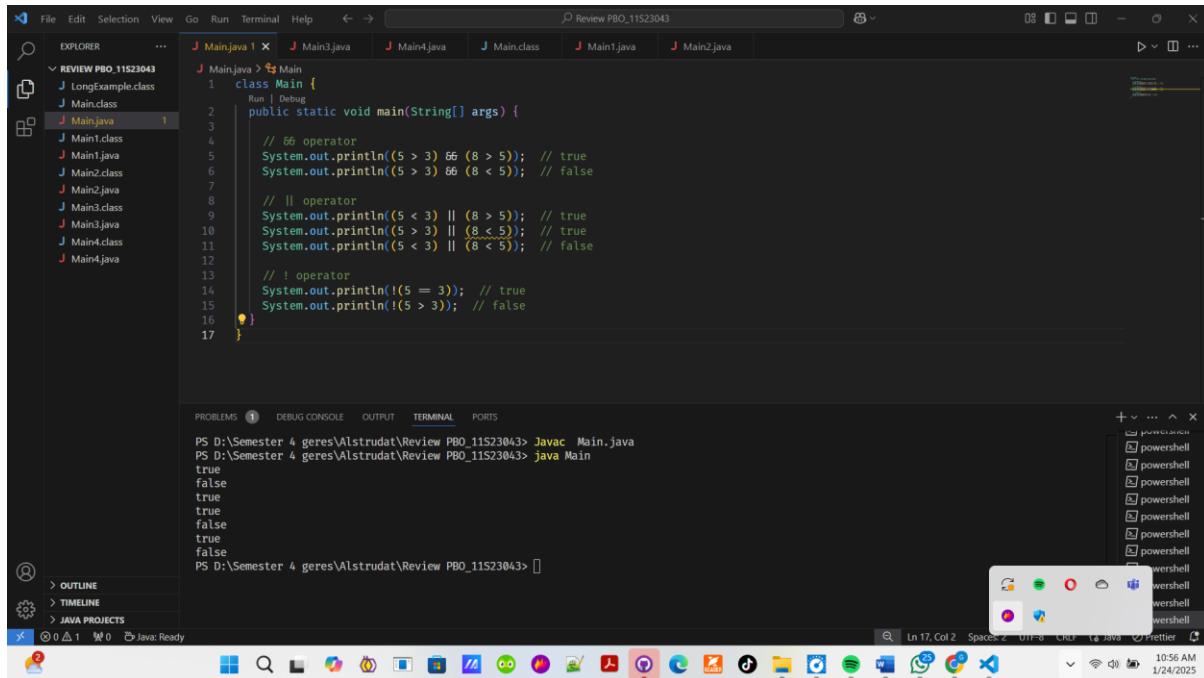
The status bar at the bottom right indicates the date and time as 10:55 AM on 1/24/2025.

# 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

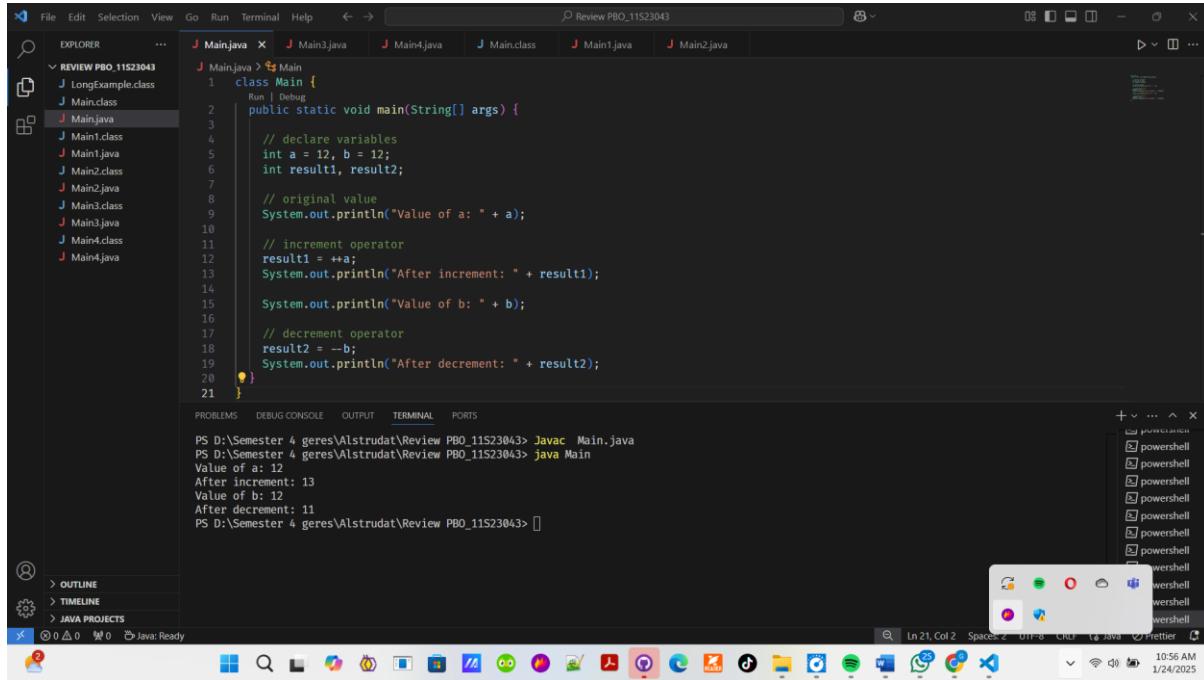
        // || operator
        System.out.println((5 < 3) || (8 > 5)); // true
        System.out.println((5 > 3) || (8 < 5)); // true
        System.out.println((5 < 3) || (8 < 5)); // false

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

The terminal below the editor 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
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 the editor 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

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

- File Explorer:** Shows multiple Java files: Main.java, Main3.java, Main4.java, Main.class, Main1.java, Main2.java, Main1.class, Main2.class, Main3.class, Main4.class, Main.java, LongExample.class.
- Code Editor:** Displays the Main.java file with the following code:

```
1 class Main {  
2     public static void main(String[] args) {  
3         String str = "Programiz";  
4         boolean result;  
5  
6         // checks if str is an instance of  
7         // the String class  
8         result = str instanceof String;  
9         System.out.println("Is str an object of String? " + result);  
10    }  
11 }
```
- 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  
Is str an object of String? true  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Status Bar:** Shows Java: Ready, along with other system icons.

### Java Ternary Operator

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

- File Explorer:** Shows multiple Java files: Main.java, Java.java, Main3.java, Main4.java, Main.class, Main1.java, Main2.java, Main1.class, Main2.class, Main3.class, Main4.class, Main.java, LongExample.class.
- Code Editor:** Displays the Java.java file with the following code:

```
1 class Java {  
2     public static void main(String[] args) {  
3         int februaryDays = 29;  
4         String result;  
5  
6         // ternary operator  
7         result = (februaryDays == 28) ? "Not a leap year" : "Leap year";  
8         System.out.println(result);  
9     }  
10 }  
11 }
```
- Terminal:** Shows the command line output:

```
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>
```
- Bottom Status Bar:** Shows Java: Ready, along with other system icons.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Challenge

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

- File Explorer:** Shows a project named "REVIEW PBO\_11S23043" containing files: Main.java, Challenge.java, Java.java, Main3.java, Main4.java, Main.class, Main1.java, Main2.java, Main3.class, Main4.class, Main5.java, Main6.java, Main7.java, Main8.java, Main9.java, Main10.java.
- Code Editor:** 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));
    }
}
```
- Terminal:** 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>
```
- Bottom Status Bar:** Shows Java: Ready, Ln 1, Col 14 (9 selected), Spaces: 2, 011-8, CPU, 11:09 AM, 1/24/2025.

### ✓ Java Basic Input and Output

#### Java Output

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

- File Explorer:** Shows a project named "REVIEW PBO\_11S23043" containing files: Main.java, Challenge.java, AssignmentOperator.java, Java.java, Main3.java, Main4.java, Main.class, Main1.java, Main2.java, Main3.class, Main4.class, Main5.java, Main6.java, Main7.java, Main8.java, Main9.java, Main10.java.
- Code Editor:** Displays the "AssignmentOperator.java" file with the following code:

```
class AssignmentOperator {
    public static void main(String[] args) {
        System.out.println("Java programming is interesting.");
    }
}
```
- Terminal:** 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>
```
- Bottom Status Bar:** Shows Java: Ready, Ln 5, Col 4, Spaces: 2, 011-8, CPU, 11:07 AM, 1/24/2025.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



print() and println()

```
1 class Output {
2     public static void main(String[] args) {
3         System.out.println("1. print");
4         System.out.println("2. print");
5         System.out.print("1. print");
6         System.out.print("2. print");
7     }
8 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> Javac Output.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> java Output  
1. print  
2. print  
1. print 2. print  
PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> [ ]

OUTLINE TIMELINE JAVA PROJECTS

Java: Ready

Ln 10, Col 2 Spaces: 2 0/11-5 CWD: 1 Java 11/11 AM 1/24/2025

## Printing Variables and Literals

```
1 class Variables {
2     public static void main(String[] args) {
3         Double number = -10.6;
4         System.out.println(x:5);
5         System.out.println(number);
6     }
7 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

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> [ ]

OUTLINE TIMELINE JAVA PROJECTS

Java: Ready

Ln 1, Col 16 (9 selected) Spaces: 2 0/11-5 CWD: 1 Java 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 title bar "Review PBO\_11S23043". The Explorer sidebar lists several Java files. The main editor window contains 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 }
```

The terminal below shows the output of running the code:

```
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>
```

### Java Input

#### Get Integer Input From the User

The screenshot shows the Visual Studio Code interface with the title bar "Review PBO\_11S23043". The Explorer sidebar lists several Java files. The main editor window contains 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 }
```

The terminal below shows the output of running the code:

```
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>
```

# **11S2215 - Algorithms and Data Structures**

## Laporan Praktikum

## Get float, double and String Input

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

- File Explorer (left):** Shows files in the current workspace, including `Input.java`, `Challenge.java`, `AssignmentOperator.java`, `Output.java`, `Variables.java`, `PrintVariables.java`, `Input.java` (selected), `Java.java`, and `Main3.java`.
- Code Editor (center):** Displays the `Input.java` source code. The code reads user input for float, double, and string values and prints them back.
- Terminal (bottom):** Shows the command-line output of running the `Input.java` program. It prompts for a float (123.4), a double (-123.4), and a string ("Haloo,I'm grace!"). The output matches the expected results.
- Bottom Status Bar:** Shows "Java: Ready".

## ✓ Java Expressions, Statements and Blocks

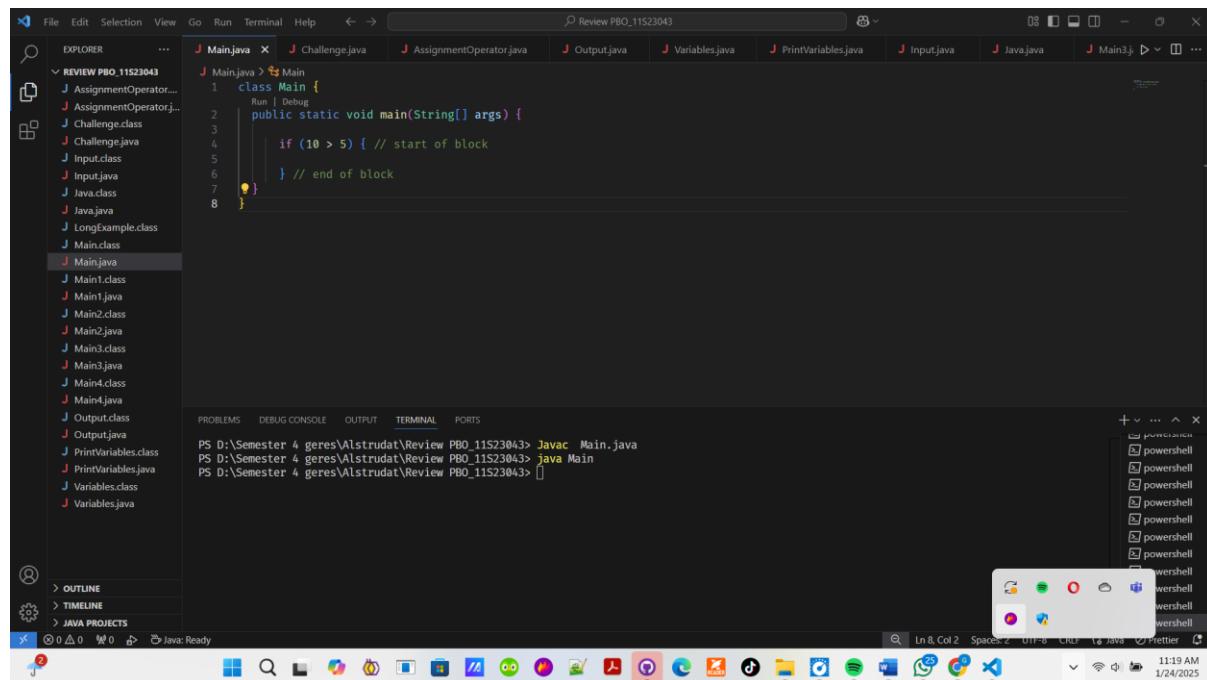
## Java Blocks

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

- File Explorer:** Shows a project named "REVIEW PBO\_11S23043" containing multiple Java files: Main.java, Challenge.java, AssignmentOperator.java, Output.java, Variables.java, PrintVariables.java, Input.java, Java.java, and Main3.java.
- Code Editor:** The main editor window displays the "Main.java" file. The code prints "Hey Grace!" when run. A yellow warning icon is present on line 8.
- Terminal:** The terminal window shows the command "javac Main.java" followed by the output "PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> java Main Hey Grace!".
- Status Bar:** The status bar at the bottom indicates "Java: Ready".

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



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

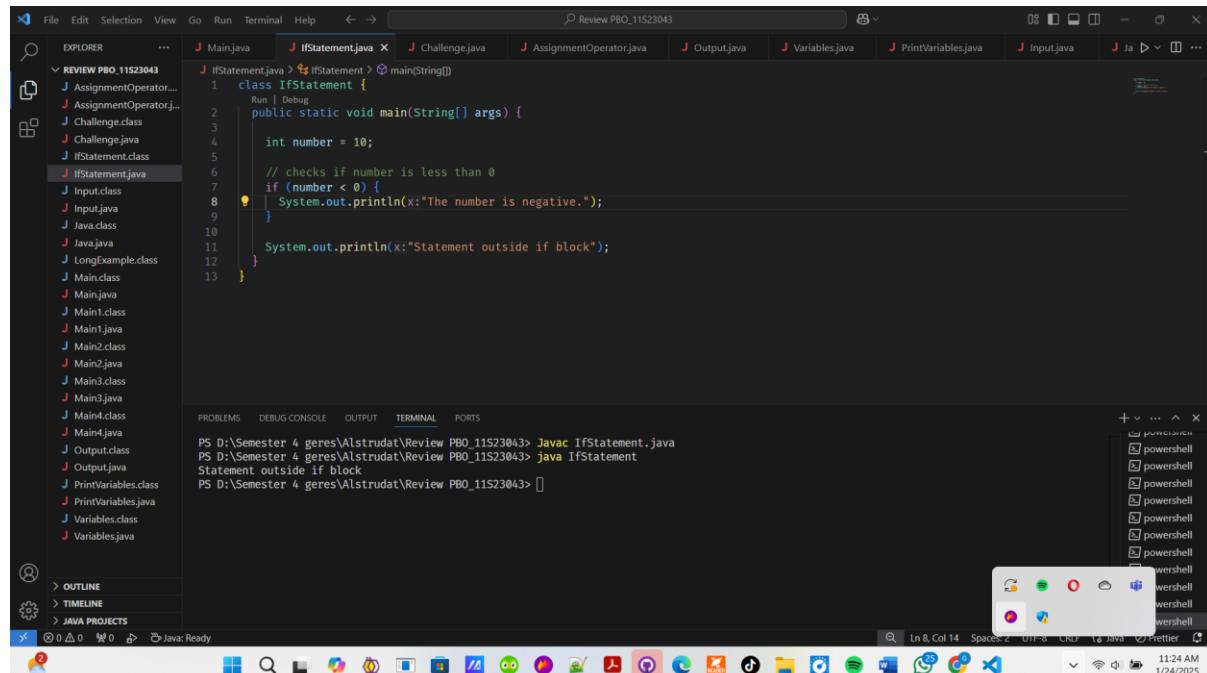
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3
4         if (10 > 5) { // start of block
5
6             } // end of block
7
8 }

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

Ln 8, Col 2 Spaces: 2 011-5 CTR L JAVA Fetter
Java: Ready
Windows Taskbar: Ln 8, Col 2 Spaces: 2 011-5 CTR L JAVA Fetter
11:19 AM 1/24/2025
```

- **Java Flow Control**
- ✓ **Java if (if-then) Statement**

### Java if Statement



```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER ... J Main.java J IfStatement.java J Challenge.java J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Java.java

J IfStatement.java > IfStatement > main(String[])
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(x:"The number is negative.");
9         }
10
11         System.out.println(x:"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> [REDACTED]

Ln 8, Col 14 Spaces: 2 011-5 CTR L JAVA Fetter
Java: Ready
Windows Taskbar: Ln 8, Col 14 Spaces: 2 011-5 CTR L JAVA Fetter
11:24 AM 1/24/2025
```

# 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 }
13 }
```

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 code editor in VS Code. The code in the main editor window is:

```
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 code is part of a project named "REVIEW PBO\_11S23043". The terminal below shows the command line 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...else Statement

The screenshot shows a Java code editor in VS Code. The code in the main editor window is:

```
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 }
```

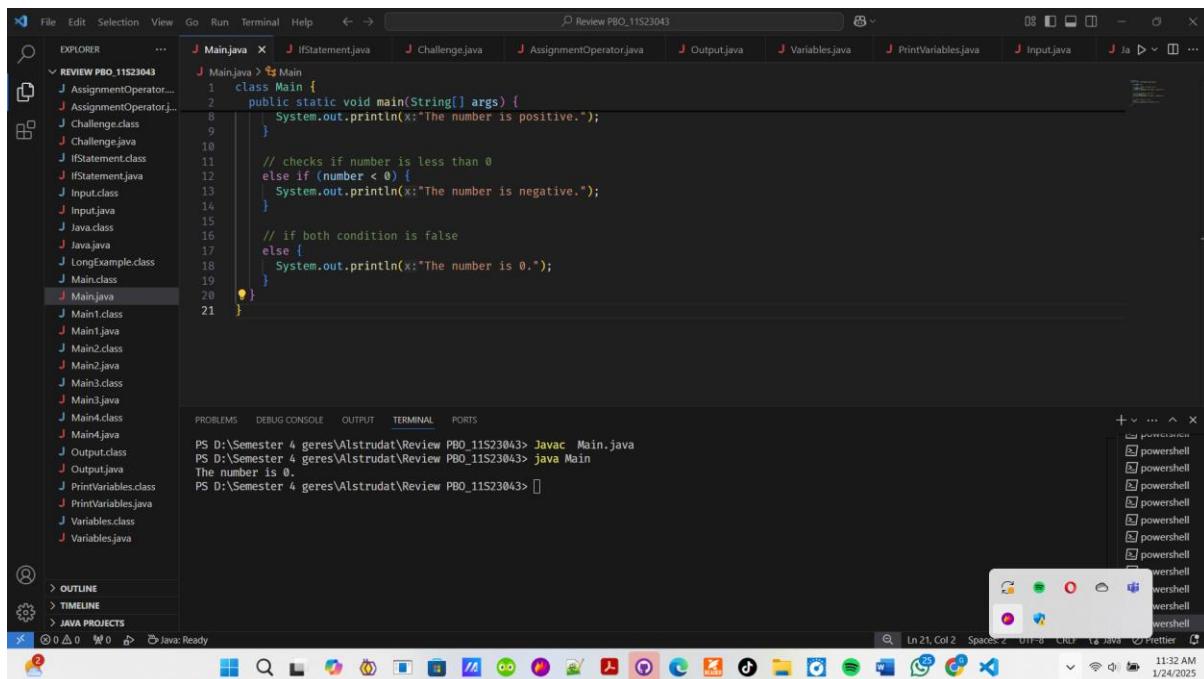
The code is part of a project named "REVIEW PBO\_11S23043". The terminal below shows the command line output of running the code:

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

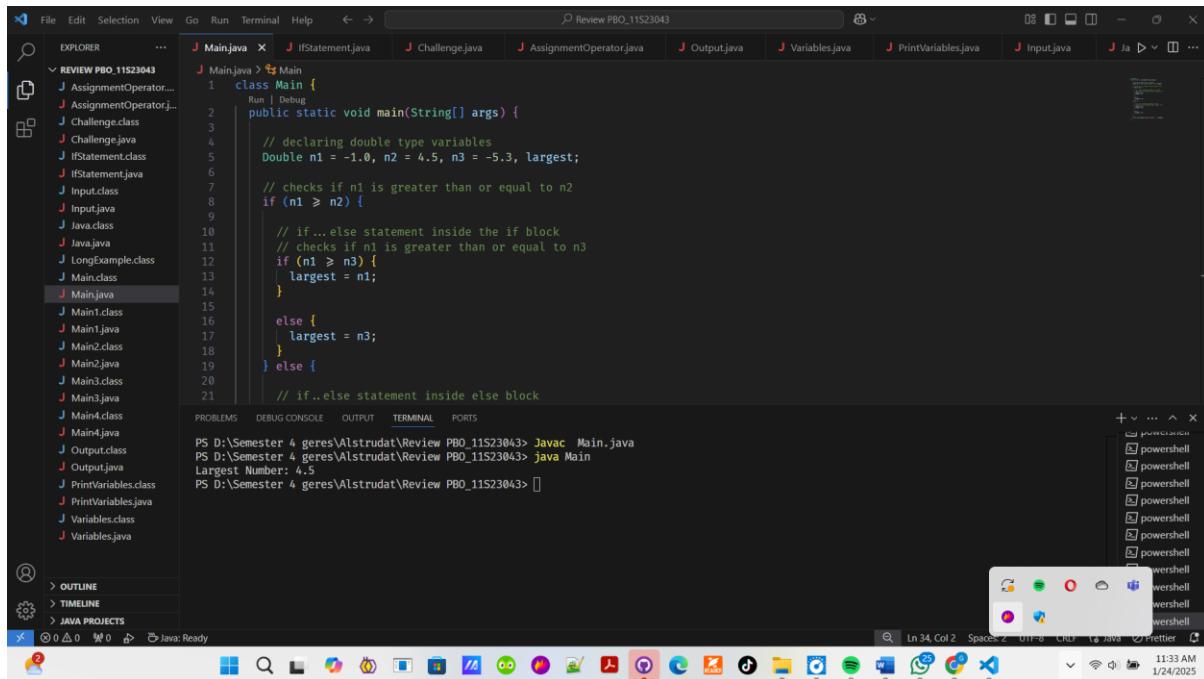
### Java if...else...if Statement



```
File Edit Selection View Go Run Terminal Help ← → ⌘ Review PBO_11S23043 J Main.java X J IfStatement.java J Challengejava J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja D ... EXPLORER J Main.java > Main 1 class Main { 2     public static void main(String[] args) { 3         System.out.println("The number is positive."); 4     } 5     // checks if number is less than 0 6     else if (number < 0) { 7         System.out.println("The number is negative."); 8     } 9     // if both condition is false 10    else { 11        System.out.println("The number is 0."); 12    } 13 } 14 15 16 17 18 19 20 21 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 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 J Main.java X J IfStatement.java J Challengejava J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja D ... EXPLORER J Main.java > Main 1 class Main { 2     public static void main(String[] args) { 3         // declaring double type variables 4         Double n1 = -1.0, n2 = 4.5, n3 = -5.3, largest; 5         // checks if n1 is greater than or equal to n2 6         if (n1 ≥ n2) { 7             // if...else statement inside the if block 8             // checks if n1 is greater than or equal to n3 9             if (n1 ≥ n3) { 10                 largest = n1; 11             } 12             else { 13                 largest = n3; 14             } 15             // if..else statement inside else block 16         } 17     } 18 } 19 20 21 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 11:32 AM 1/24/2025
```

### Challenge

```

File Edit Selection View Go Run Terminal Help ← → Review PBO_11S23043
EXPLORER J Main.java J IfStatement.java J PassOrFail.java X J AssignmentOperator.java J Output.java J Variables.java J PrintVariables.java J Input.java J Ja ▾ ▯ ...
J AssignmentOperator...
J AssignmentOperator.j...
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
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>

```

This screenshot shows a Java application using a ternary operator to determine if a score is pass or fail. The code is as follows:

```

public class PassOrFail {
    public static String checkPassOrFail(int score) {
        return (score >= 50) ? "Pass" : "Fail";
    }

    public static void main(String[] args) {
        int score = 55; // Example score
        System.out.println("Result: " + checkPassOrFail(score)); // Output: Pass
    }
}

```

### ✓ Java Ternary Operator

```

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 Variables.java J PrintVariables.java J Input.java J Ja ▾ ▯ ...
J Main.java
J AssignmentOperator...
J AssignmentOperator.j...
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
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
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>

```

This screenshot shows a Java application that takes user input for marks and uses a ternary operator to determine if the student has passed or failed. The code is as follows:

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        // take input from users
        Scanner input = new Scanner(System.in);
        System.out.println("Enter your marks: ");
        double marks = input.nextDouble();

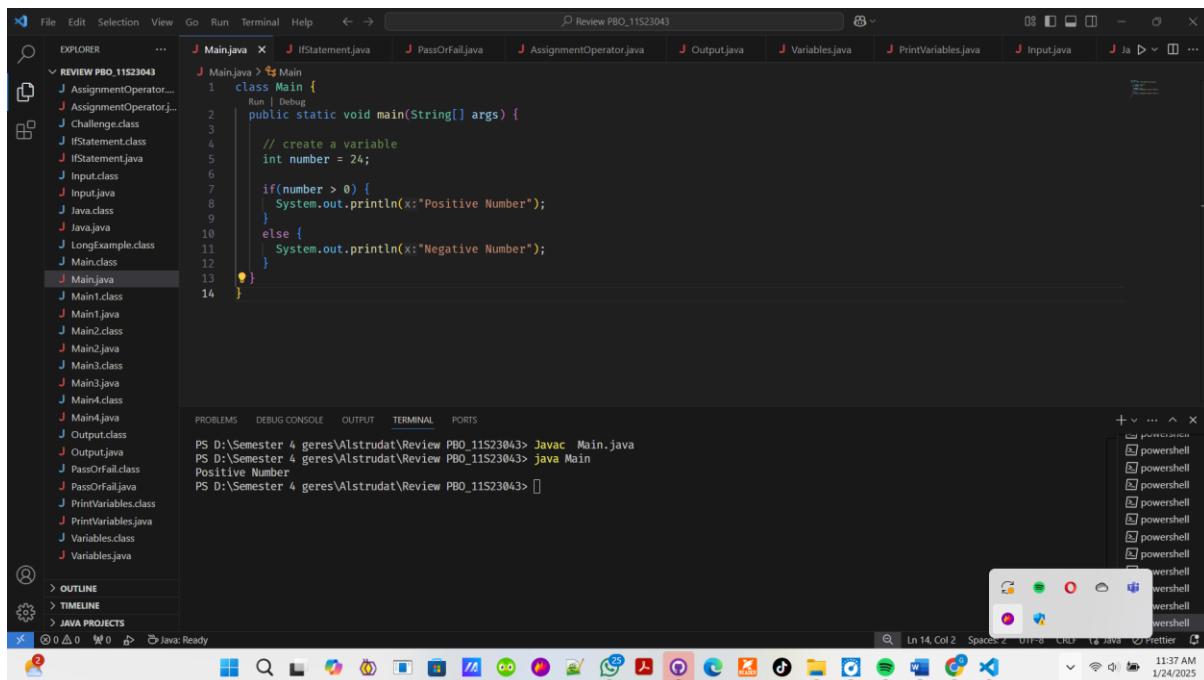
        // ternary operator checks if
        // marks is greater than 40
        String result = (marks > 40) ? "pass" : "fail";

        System.out.println("You " + result + " the exam.");
        input.close();
    }
}

```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

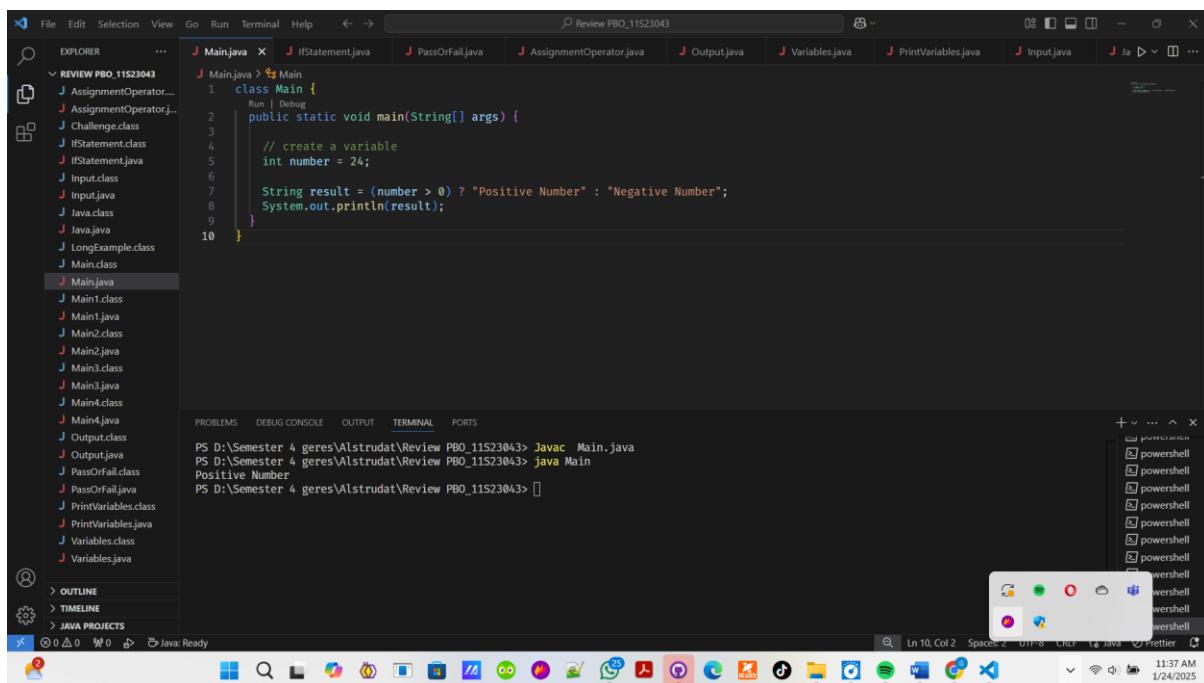


The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays a Java file named Main.java. The code contains an if-statement that prints "Positive Number" if the variable number is greater than 0, and "Negative Number" otherwise. The terminal below the editor shows the command line output of running the Java compiler (javac) and the resulting Java program (java Main), which correctly prints "Positive Number". The taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as 11:37 AM on 1/24/2025.

```
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> [ ]
```



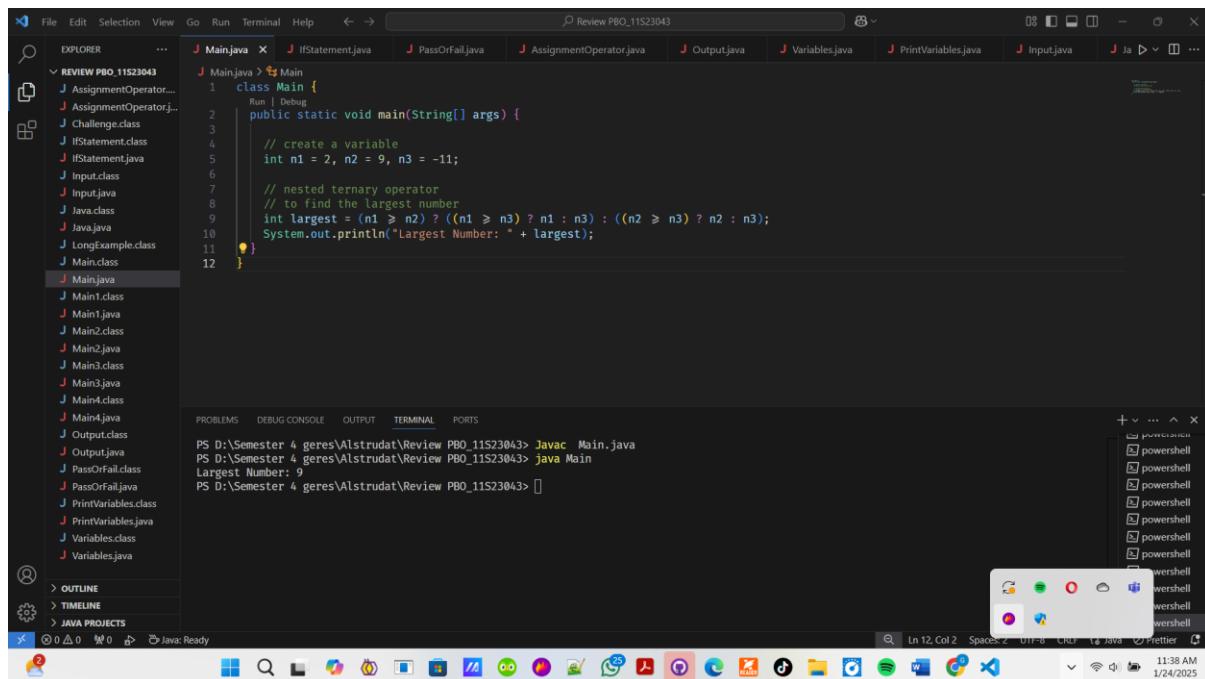
The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays a Java file named Main.java. The code uses a ternary operator to determine the string result based on the value of the variable number. The terminal below the editor shows the command line output of running the Java compiler (javac) and the resulting Java program (java Main), which correctly prints "Positive Number". The taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as 11:37 AM on 1/24/2025.

```
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> [ ]
```

### Nested Ternary Operators



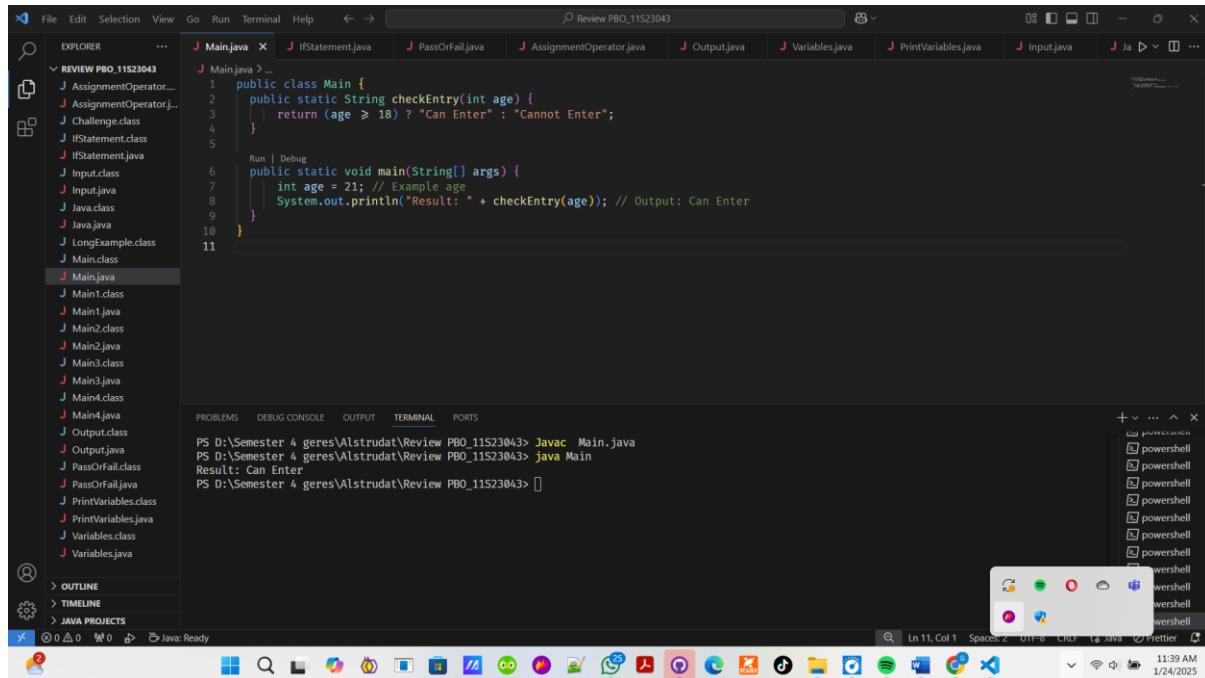
The screenshot shows a Java file named Main.java in the VS Code editor. The code contains a Main class with a main method that declares three integer variables n1, n2, and n3, initializes them, and then uses a nested ternary operator to find the largest number among them. The terminal tab shows the command-line output of running the Java compiler (javac) and the Java interpreter (java) on the Main class, resulting in the output "Largest Number: 9".

```
1 class Main {
2     public static void main(String[] args) {
3         // create a variable
4         int n1 = 2, n2 = 9, n3 = -11;
5
6         // nested ternary operator
7         // to find the largest number
8         int largest = (n1 >= n2) ? ((n1 >= n3) ? n1 : n3) : ((n2 >= n3) ? n2 : n3);
9         System.out.println("Largest Number: " + largest);
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
Largest Number: 9
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 Main class with a checkEntry method that takes an integer age as input and returns a string indicating whether the person can enter or not. The main method calls this function with an example age of 21 and prints the result. The terminal tab shows the command-line output of running the Java compiler (javac) and the Java interpreter (java) on the Main class, resulting in the output "Result: Can Enter".

```
1 public class Main {
2     public static String checkEntry(int age) {
3         return (age >= 18) ? "Can Enter" : "Cannot Enter";
4     }
5
6     public static void main(String[] args) {
7         int age = 21; // Example age
8         System.out.println("Result: " + checkEntry(age)); // Output: Can Enter
9     }
10 }
```

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
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

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

- File Explorer:** Shows multiple Java files including Main.java, IfStatement.java, PassOrFail.java, AssignmentOperator.java, Output.java, Variables.java, PrintVariables.java, and Input.java.
- Code Editor:** The Main.java file is open, containing the following code:

```
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 }
```
- Terminal:** The terminal window shows the 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
Java is fun
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Taskbar:** The taskbar at the bottom shows various pinned icons, including Microsoft Edge, File Explorer, and File History.

Display numbers from 1 to 5

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

- File Explorer:** Shows multiple Java files including Main.java, IfStatement.java, PassOrFail.java, AssignmentOperator.java, Output.java, Variables.java, PrintVariables.java, and Input.java.
- Code Editor:** The Main.java file is open, containing the following code:

```
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 }
```
- Terminal:** The terminal window shows the 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
1
2
3
4
5
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Taskbar:** The taskbar at the bottom shows various pinned icons, including Microsoft Edge, File Explorer, and File History.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Display Sum of n Natural Numbers

The screenshot shows a Java application named Main.java. The code initializes a variable sum to 0 and n to 1000. It then uses a for loop to iterate from 1 to n, adding each value to sum. Finally, it prints the result. 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 the same Java application Main.java. The code is identical to the one in the previous screenshot. 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

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

- File Explorer:** Shows a tree view of Java files under the project "REVIEW PBO\_11S23043". The file "Main.java" is selected.
- Code Editor:** Displays the content of Main.java:

```
class Main {
    public static void main(String[] args) {
        int[] numbers = {3, 7, 5, -5};
        for (int number : numbers) {
            System.out.println(number);
        }
    }
}
```
- Terminal:** Shows the command-line interface output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> Java Main
3
7
5
-5
```
- Bottom Bar:** Includes icons for file operations, terminal, and Java-related tools like prettier and powershell.

## Java Infinite for Loop

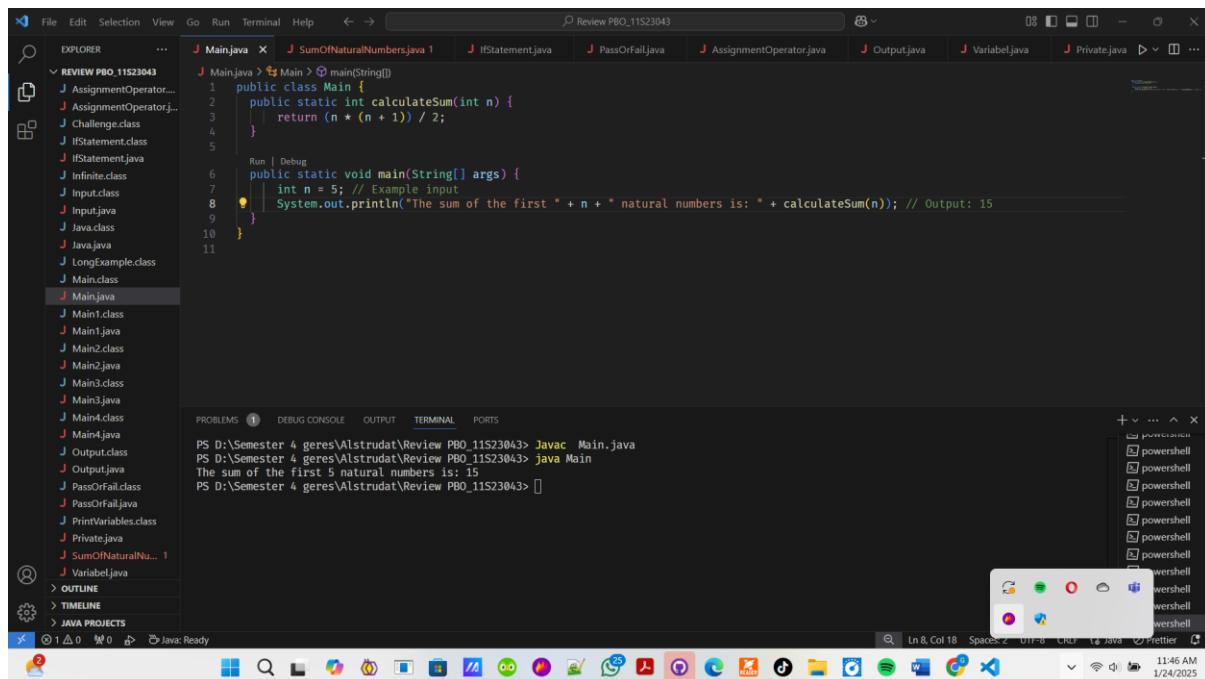
The screenshot shows a Java code editor interface with the following details:

- File Explorer:** Shows a tree view of Java files under the project "REVIEW PBO\_11523043". The file "Infinite.java" is selected.
- Code Editor:** Displays the "Infinite.java" code:

```
1 // Infinite for Loop
2
3 class Infinite {
4     Run | Debug
5     public static void main(String[] args) {
6
7         int sum = 0;
8
9         for (int i = 1; i <= 10; --i) {
10             System.out.println("Hello");
11         }
12     }
13 }
```
- Terminal:** Shows the output of the "Hello" prints:

```
Hello
```
- Bottom Bar:** Includes icons for file operations (New, Open, Save, Find, Replace, Cut, Copy, Paste), a search bar, and system status indicators (CPU, GPU, RAM, Disk, Network).

### Challenge



The screenshot shows a Java application calculating the sum of the first 5 natural numbers. The code in Main.java is:

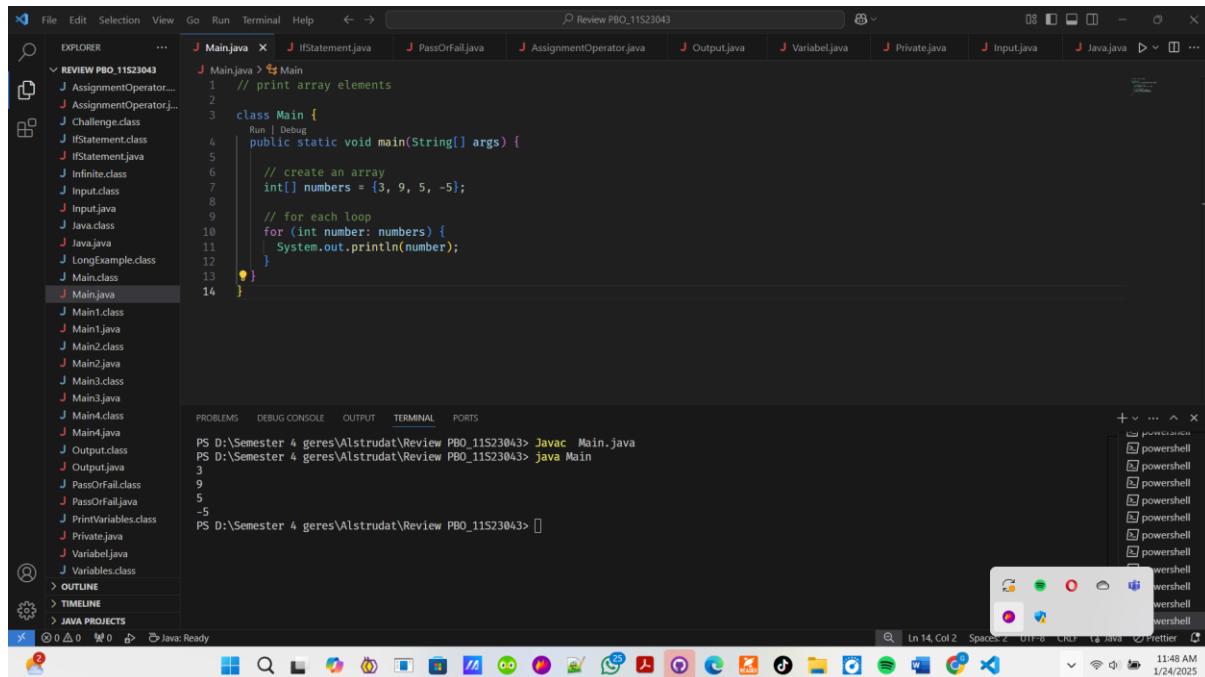
```
public class Main {
    public static int calculateSum(int n) {
        return (n * (n + 1)) / 2;
    }

    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.

### ✓ Java for-each Loop

#### Print Array Elements



The screenshot shows a Java application printing the elements of an integer array using a for-each loop. The code in Main.java is:

```
// print array elements
class Main {
    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 elements of the array.

# 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;

    // iterating through each element of the array
    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> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
Sum = 19
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

### 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:

```
char[] vowels = {'a', 'e', 'i', 'o', 'u'};

// iterating through an array using a for loop
for (int i = 0; i < vowels.length; ++ i) {
    System.out.println(vowels[i]);
}
```

The terminal output shows the result 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
e
i
o
u
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Using for-each Loop

The screenshot shows a Java code editor in VS Code. The main pane displays the following code:

```
class Main {
    public static void main(String[] args) {
        char[] vowels = {'a', 'e', 'i', 'o', 'u'};
        // iterating through an array using the for-each loop
        for (char item: vowels) {
            System.out.println(item);
        }
    }
}
```

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
a
e
i
o
u
```

### Challenge

The screenshot shows a Java code editor in VS Code. The main pane displays the following code:

```
public class Main {
    public static int calculateSum(int[] arr) {
        int sum = 0;
        for (int num : arr) {
            sum += num;
        }
        return sum;
    }

    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50}; // Example array
        System.out.println("The sum of the array elements is: " + calculateSum(arr)); // Output: 150
    }
}
```

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
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 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 IfStatement.java J Infinite.class J Input.class J Input.java J Java.class J Java.java J LongExample.class J Main.class J Main.java 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
J Challenge.class J IfStatement.class J Infinite.class J Input.class J Input.java J Java.class J Java.java J LongExample.class J Main.class J Main.java 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
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>

```

This screenshot shows a Java application in VS Code. The code in the main file `Main.java` is as follows:

```

class Main {
    public static void main(String[] args) {
        int i = 1, n = 5;
        while(i <= n) {
            System.out.println(i);
            i++;
        }
    }
}

```

The terminal window shows the output of the program: "1", "2", "3", "4", and "5".

Sum of Positive Numbers Only

```

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 Variabel.java J Private.java J Input.java J Java.java
J AssignmentOperator... J IfStatement.class J IfStatement.java J Infinite.class J Input.class J Input.java J Java.class J Java.java J LongExample.class J Main.class J Main.java 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
J Challenge.class J IfStatement.class J Infinite.class J Input.class J Input.java J Java.class J Java.java J LongExample.class J Main.class J Main.java 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
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>

```

This screenshot shows a Java application in VS Code. The code in the main file `Main.java` is as follows:

```

class Main {
    public static void main(String[] args) {
        System.out.println("Enter a number");
        int number = input.nextInt();
        // while loop continues
        // until entered number is positive
        while (number >= 0) {
            // add only positive numbers
            sum += number;

            System.out.println("Enter a number");
            number = input.nextInt();
        }

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

```

The terminal window shows the user entering numbers and the program calculating their sum: "4", "5", "6", "7", "0", and "-1". The final output is "Sum = 28".

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Java do...while loop

Display Numbers from 1 to 5

```
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     public static void main(String[] args) {
9
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 }
```

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> 
```

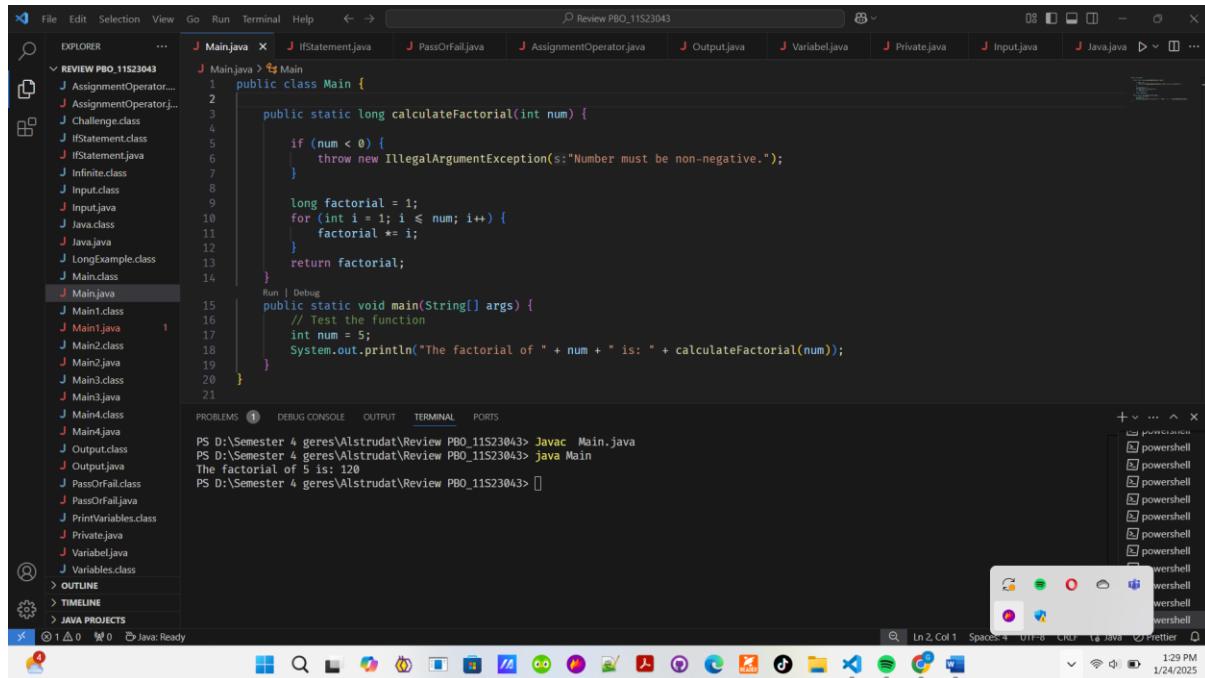
### Sum of Positive Numbers

```
1 // Java program to find the sum of positive numbers
2
3 import java.util.Scanner;
4
5 class Main {
6     public static void main(String[] args) {
7
8         int sum = 0;
9         int number = 0;
10
11         // create an object of Scanner class
12         Scanner input = new Scanner(System.in);
13
14         // do...while loop continues
15         // until entered number is positive
16         do {
17             // add only positive numbers
18             sum += number;
19             System.out.println("Enter a number");
20             number = input.nextInt();
21         } while(number >= 0);
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
Enter a number
3
Enter a number
-1
Sum = 3
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> java Main
Enter a number
-7
Sum = 0
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> 
```

### Challenge



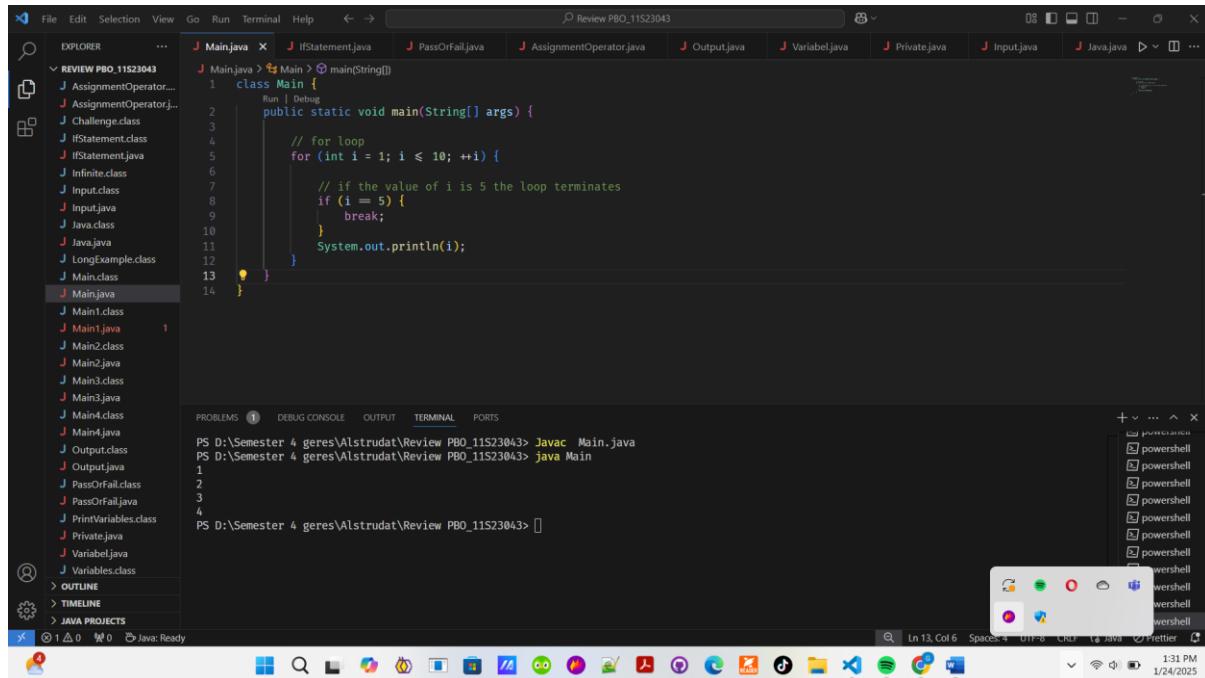
The screenshot shows a Java application named Main.java. The code defines a public class Main with a main method. The main method contains a for loop that calculates the factorial of a given number. The output of the program is displayed in the terminal.

```
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));
}
```

Terminal output:

```
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 break statement



The screenshot shows a Java application named Main.java. The code defines a public class Main with a main method. The main method contains a for loop that prints integers from 1 to 10. However, when the value of i reaches 5, the loop terminates early due to a break statement. The output of the program is displayed in the terminal.

```
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);
    }
}
```

Terminal output:

```
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
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### labeled break Statement

The screenshot shows a Java code editor in VS Code. The current file is LabeledBreak.java. The code contains two nested for loops. The outer loop is labeled 'first' and the inner loop is labeled 'second'. A break statement is used inside the 'second' loop to exit the 'first' loop. The code is as follows:

```
class LabeledBreak {
    public static void main(String[] args) {
        // the for loop is labeled as first
        for( int i = 1; i < 5; i++ ) {
            // the for loop is labeled as 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 terminal output shows the execution of the code, which prints the following:

```
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

The screenshot shows a Java code editor in VS Code. The current file is Main.java. The code defines a class Main with a static method isPrime that checks if a number is prime. It also has a main method that prints the result for the number 29. The code is as follows:

```
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));
    }
}
```

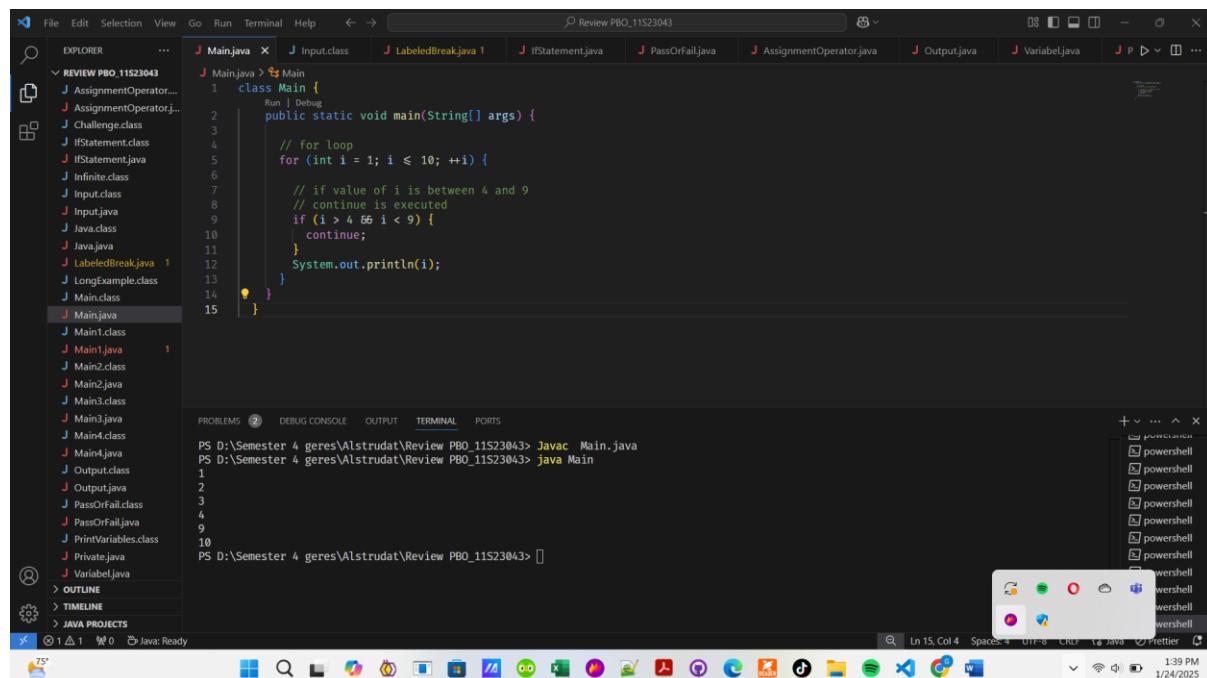
The terminal output shows the execution of the code, which prints the following:

```
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> [REDACTED]
```

### ✓ Java continue Statement

#### Java continue

Java continue statement

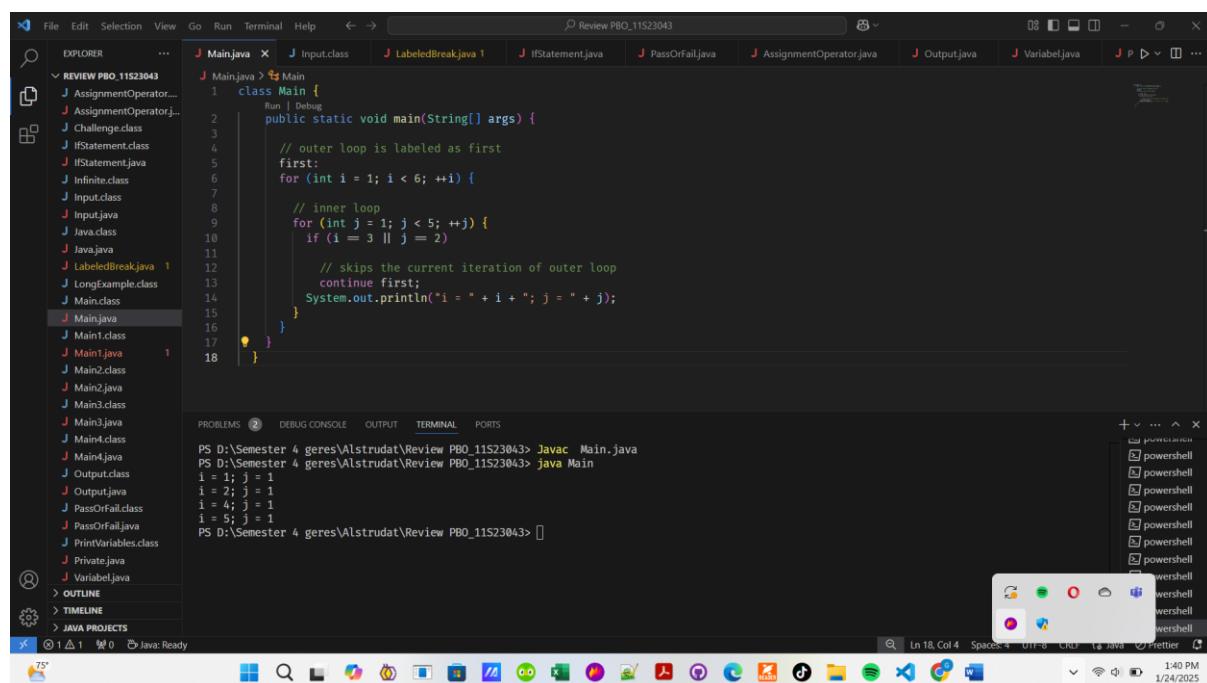


```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043
J Main.java > Main
1 class Main {
2     public static void main(String[] args) {
3         // for loop
4         for (int i = 1; i <= 10; ++i) {
5             // if value of i is between 4 and 9
6             // continue is executed
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
5
6
7
8
9
10

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

labeled continue Statement



```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043
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 = 3; j = 1
i = 4; j = 1
i = 5; j = 1

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

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java switch Statement

#### Java switch Statement

The screenshot shows a Java project named "REVIEW PBO\_11S23043" in the Explorer sidebar. The main.java file is open in the editor, displaying the following code:

```
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         int number = 44;
7         String size;
8
9         // switch statement to check size
10        switch (number) {
11            case 29:
12                size = "Small";
13                break;
14
15            case 42:
16                size = "Medium";
17                break;
18
19            default:
20                size = "Large";
21        }
22    }
23}
```

The terminal tab shows the output of running the program:

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

#### break Statement in Java switch...case

The screenshot shows a Java project named "REVIEW PBO\_11S23043" in the Explorer sidebar. The main.java file is open in the editor, displaying 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("Case 1");
9                 // matching case
10            case 2:
11                System.out.println("Case 2");
12
13            case 3:
14                System.out.println("Case 3");
15
16            default:
17                System.out.println("Default case");
18        }
19    }
20}
```

The terminal tab shows the output of running the program:

```
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
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

# 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 contains a switch statement with three cases (2, 3, and default) and a fall-through case. The output window shows the program's execution and the result 'Unknown Size'.

```
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 calculate method that takes two doubles and an operator character, then performs the corresponding arithmetic operation using a switch statement. The output window shows the result of calling the method with parameters 5.0 and 3.0.

```
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 MainJava 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 v ...
J MainJava > Main {
1 class Main {
2     public static void main(String[] args) {
3
4         // create an array
5         int[] age = {12, 4, 5, 2, 5};
6
7         // access each array elements
8         System.out.println("Accessing Elements of Array:");
9         System.out.println("First Element: " + age[0]);
10        System.out.println("Second Element: " + age[1]);
11        System.out.println("Third Element: " + age[2]);
12        System.out.println("Fourth Element: " + age[3]);
13        System.out.println("Fifth Element: " + age[4]);
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 MainJava 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 v ...
J MainJava > 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 }

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 development environment with the following details:

- File Explorer:** Shows a tree view of Java files under the project "REVIEW PBO\_11S23043". The file "Main.java" is selected.
- Code Editor:** Displays the content of "Main.java". The code uses a for-each loop to print array elements.
- Terminal:** Shows the command-line interface output for running the program. It includes:
  - Java Main
  - Using for-each Loop:
  - 12
  - 4
  - 5
- Status Bar:** Shows "Java: Ready" and other system information like battery level and signal strength.

## Challenge

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Review PBO\_11S23043.
- Left Sidebar (PROJECT EXPLORE):** REVIEW PBO\_11S23043, J AssignmentOperator..., J AssignmentOperator..., J Challenge.class, J IfStatement.class, J IfStatement.java, J Infinite.class, J Input.class, J Input.java, J Java.class, J Java.java, J LabeledBreak.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 Private.java, J Variabel.java, > OUTLINE, > TIMELINE, > JAVA PROJECTS.
- Central Area (Editor):** The Main.java file is open, displaying code to calculate the average of an array of integers. The code includes a method `calculateAverage` that takes an array of integers and returns its average. It handles the case where the array is empty by throwing an `IllegalArgumentException`.
- Bottom Bar:** PROBLEMS, DEBUG CONSOLE, OUTPUT, TERMINAL, PORTS. The TERMINAL tab is active, showing command-line output for running the Java code.
- Right Sidebar:** A list of recent files or projects named powershell.
- Taskbar:** Shows various application icons including File Explorer, Task Manager, and a browser icon.

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

# **11S2215 - Algorithms and Data Structures**

## Laporan Praktikum

✓ Java Multidimensional Arrays

## 2-dimensional Array

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

- File Explorer:** Shows files like `EXPLORER`, `REVIEW PBO_11S23043`, and numerous Java source files such as `Output.java`, `Variabel.java`, `Private.java`, `Input.java`, `MultidimensionalArray.java`, `Main.java`, `Main3.java`, `Main1.java`, and `Maintest.java`.
- Code Editor:** The `MultidimensionalArray.java` file is open, displaying Java code to print the dimensions of a 2D array.
- Terminal:** The terminal window shows the command `java MultidimensionalArray` being run, followed by the output: "Length of row 1: 3", "Length of row 2: 4", and "Length of row 3: 1".
- Bottom Bar:** Includes icons for file operations, a search bar, and system status indicators.

## 3-dimensional Array

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

- File Explorer:** Shows various Java files including Main.java, ThreeArray.java, Main.class, Main.class, Input.class, LabeledBreak.java, IfStatement.java, PassOrFail.java, and AssignmentOperator.java.
- Code Editor:** The ThreeArray.java file is open, displaying code to create a 3D array and iterate through its elements using nested for-each loops.
- Terminal:** The terminal window shows the command being run: `java ThreeArray`.
- Output:** The terminal output shows the elements of the 3D array being printed to the console.
- Bottom Status Bar:** Shows Java: Ready, 0 errors, 0 warnings, and 0 info messages.
- System Tray:** Shows icons for power shell, task manager, and other system utilities.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java Copy Arrays

#### Copying Arrays Using Assignment Operator

The screenshot shows a Java IDE interface with a project named "REVIEW PBO\_11S23043". The "Main.java" file is open in the editor, containing the following code:

```
1 class Main {
2     public static void main(String[] args) {
3         int [] numbers = {1, 2, 3, 4, 5, 6};
4         int [] positiveNumbers = numbers; // copying arrays
5
6         for (int number: positiveNumbers) {
7             System.out.print(number + " ");
8         }
9     }
10}
11}
```

The "TERMINAL" tab 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
1, 2, 3, 4, 5, 6,
```

The taskbar at the bottom shows various application icons.

#### Copying 2d Arrays using arraycopy()

The screenshot shows a Java IDE interface with a project named "REVIEW PBO\_11S23043". The "Main.java" file is open in the editor, containing the following code:

```
1 import java.util.Arrays;
2
3 class Main {
4     public static void main(String[] args) {
5         int[][] source = {
6             {1, 2, 3, 4},
7             {5, 6},
8             {0, 2, 42, -4, 5}
9         };
10
11         int[][] destination = new int[source.length][];
12
13         for (int i = 0; i < source.length; ++i) {
14             // allocating space for each row of destination array
15             destination[i] = new int[source[i].length];
16             System.arraycopy(source[i], 0, destination[i], 0, source[i].length);
17         }
18
19         // displaying destination array
20     }
21 }
```

The "TERMINAL" tab 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
[[1, 2, 3, 4], [5, 6], [0, 2, 42, -4, 5]]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

The taskbar at the bottom shows various application icons.

✓ Java Class and Objects

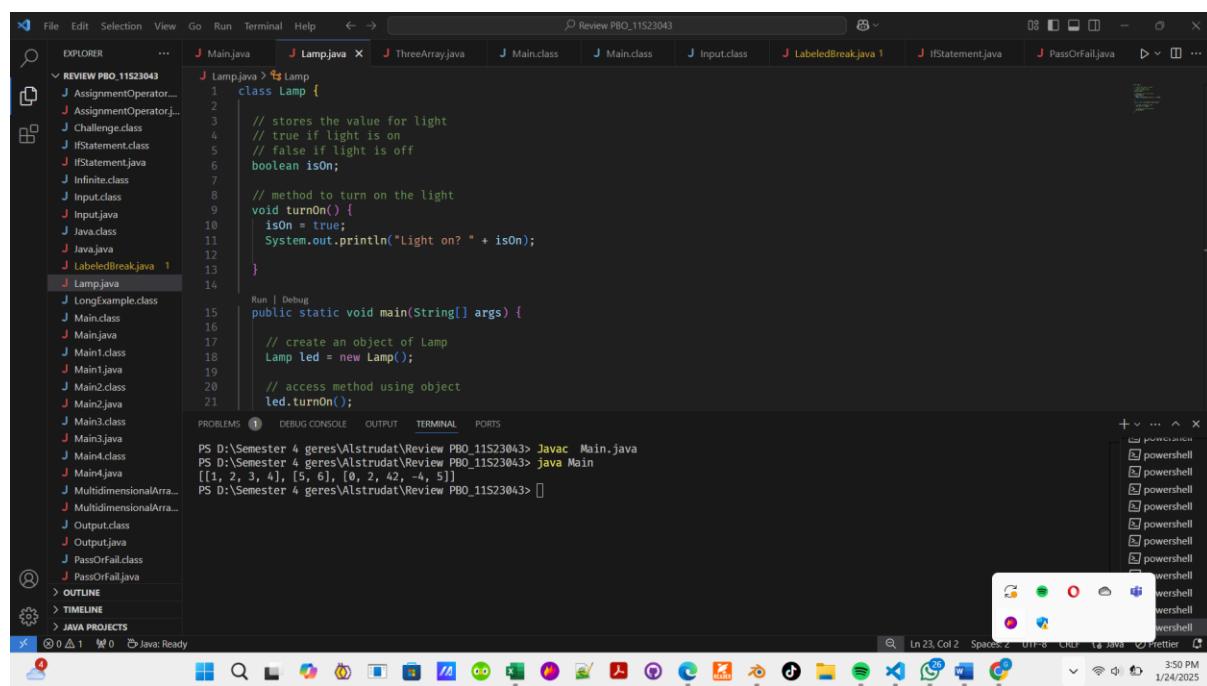
Java Class

```
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 Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Standard toolbar icons.
- Left Sidebar (EXPLORER):** Shows a tree view of files under "REVIEW PBO\_11S23043".
- Central Editor:** Displays the content of "Lamp.java".
- Bottom Status Bar:** Shows "Java: Ready".
- Bottom Taskbar:** Shows various application icons.
- Bottom Right Corner:** Shows system status icons like battery level, signal strength, and date/time.

```
Lamp.java  
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  
15 public static void main(String[] args) {  
16     // create an object of Lamp  
17     Lamp led = new Lamp();  
18  
19     // access method using object  
20     led.turnOn();  
21 }
```

Terminal Output (PROBLEMS tab):

```
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>
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### Java Methods

The screenshot shows a Java file named Main.java in the VS Code editor. The code defines a class Main with a main method and an addNumbers method. The output window shows the execution of the program, printing the sum of two integers.

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

    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);
    }
}
```

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 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 file named Main.java in the VS Code editor. The code defines a class Main with two display methods (one taking one parameter, one taking two parameters) and a main method that calls both. The output window shows the execution of the program, printing the arguments passed.

```
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);
    }

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

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
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

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

- File Explorer:** Shows files like Main.java, Lamp.java, ThreeArray.java, Main.class, Main.class, Input.class, LabeledBreak.java, IfStatement.java, PassOrFail.java, AssignmentOperator.java, AssignmentOperatorOr.java, Challenge.class, IfStatement.class, Infinite.class, Input.class, Input.java, Java.class, Javajava, LabeledBreak.java, Lamp.java, LongExample.class, Main.class, Main.java (selected), Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, MultidimensionalArra..., MultidimensionalArra..., Output.class, Output.java, PassOrFail.class, PassOrFail.java, OUTLINE, TIMELINE, and JAVA PROJECTS.
- Code Editor:** The Main.java file is open, containing the following code:

```
String language;
// constructor with no parameter
Main() {
    this.language = "Java";
}
// constructor with a single parameter
Main(String language) {
    this.language = language;
}

public void getName() {
    System.out.println("Programming Language: " + this.language);
}

Run | Debug
public static void main(String[] args) {
    // call constructor with no parameter
}
```
- Terminal:** The terminal window shows the following output:

```
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>
```
- Bottom Bar:** Includes icons for file operations, search, and system status.

✓ Java Static Keyword

## Use of static block in java

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

- File Explorer:** Shows various Java files including Main.java, Lamp.java, ThreeArray.java, Main.class, Main.class, Input.class, Input.java, Java.class, Java.java, LabeledBreak.java, Lamp.java, LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, MultidimensionalArra..., MultidimensionalArra..., Second Static block, a = 23, b = 92, max = 30.
- Terminal:** Displays the command "javac Main.java" followed by the output:

```
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
```
- Status Bar:** Shows Java: Ready, Ln 30, Col 3, Spaces: 4, UTF-8, CR/LF, Java, Fretter, and system icons for battery, signal, and volume.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java Strings

Create a String in Java

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a Java file named Main.java with the following content:

```
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 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
Java
Python
JavaScript
```

### Challenge

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The code editor displays a Java file named Main.java with the following content:

```
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 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
Test 1: String isn't empty
Test 2: String is empty
Test 3: String is empty
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java this Keyword

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> [ ]
```

Java: Ready

### ✓ Java final keyword

#### Java final Variable

Main.java

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

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> [ ]
```

Java: Ready

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java Recursion

#### Factorial of a Number Using Recursion

The screenshot shows a Microsoft Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays a Java file named Main.java. The code defines a class Main with a static method factorial that calculates the factorial of a number using recursion. The terminal tab shows the command line output of running the Java compiler (javac) and the Java runtime (java) on the Main.java file, resulting in the factorial value for n=4.

```
class Main {
    static int factorial( int n ) {
        if (n != 0) // termination condition
            return n * factorial(n-1); // recursive call
        else
            return 1;
    }

    public static void main(String[] args) {
        int number = 4, result;
        result = factorial(number);
        System.out.println(number + " factorial = " + result);
    }
}
```

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 Microsoft Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays a Java file named Main.java. The code defines a class Main with a static long method calculateFactorial that calculates the factorial of a non-negative integer. The terminal tab shows the command line output of running the Java compiler (javac) and the Java runtime (java) on the Main.java file, resulting in the factorial value for num=5.

```
public class Main {
    // Method to calculate the factorial of a number
    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; // Multiply factorial by the current number
        }
        return factorial;
    }

    public static void main(String[] args) {
        // Example usage
        int num = 5; // Test input
        try {
            System.out.println("The factorial of " + num + " is: " + calculateFactorial(num));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

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>
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### • Java OOP (II)

#### Java Inheritance

A screenshot of a Java IDE (IntelliJ IDEA) showing a project named "REVIEW PBO\_11S23043". The code editor displays a Main.java file with the following content:

```
class Animal {
    // field and method of the parent class
    String name;
    public void eat() {
        System.out.println("I can eat");
    }
}

// inherit from Animal
class Dog extends Animal {
    // new method in subclass
    public void display() {
        System.out.println("My name is " + name);
    }
}

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

The terminal window shows the 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
My name is Rohu
I can eat
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```

#### Method Overriding

A screenshot of a Java IDE (IntelliJ IDEA) showing a project named "REVIEW PBO\_11S23043". The code editor displays a Main.java file with the following content:

```
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) {
        Dog d1 = new Dog();
        d1.displayInfo();
    }
}
```

The terminal window shows the 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
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 a Java code editor interface 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 Assignment.java.
- Code Editor:** Displays the Main.java file with the following code:

```
>Main.java > Dog > display()
1 class Animal {
2
3     // overridden method
4     public void display(){
5         System.out.println("I am an animal");
6     }
7 }
8
9 class Dog extends Animal {
10
11    // overriding method
12    @Override
13    public void display(){
14        System.out.println("I am a dog");
15    }
16
17    public void printMessage(){
18        display();
19    }
20 }
21
22 class Main {
```
- 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
I am a dog
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Bar:** Includes icons for file operations, search, navigation, and system status.

✓ Java Abstract Class and Abstract Methods

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

- File Explorer:** Shows files like Input.class, Input.java, Java.class, Java.java, LabeledBreak.java, Lamp.java, Language.class, LongExample.class, Main.class, Main.java, Main1.class, Main2.class, Main3.class, Main4.class, Main5.java, MultidimensionalArr..., Output.class, Output.java, PassOrFail.class, PassOrFail.java, PrintVariables.class, Private.java, ThreeArray.class, ThreeArray.java, Variabel.java.
- Editor:** The Main.java file is open, containing the following code:

```
abstract class Language {  
    // method of abstract class  
    public void display() {  
        System.out.println("This is Java Programming");  
    }  
}  
class Main extends Language {  
    public static void main(String[] args) {  
        // create an object of Main  
        Main obj = new Main();  
        // access method of abstract class  
        // using object of Main class  
        obj.display();  
    }  
}
```
- Terminal:** The terminal window 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  
This is Java Programming  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom Bar:** Shows icons for browser, search, file explorer, taskbar, and system status.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java Interface

Main.java

```
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: Ready

### ✓ Java Polymorphism

Main.java

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

Java: Ready

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum



### ✓ Java Encapsulation

The screenshot shows the Visual Studio Code interface with the title bar "Review PBO\_11S23043". The left sidebar shows a project tree with files like Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and Assignment. The main editor window displays the following Java code:

```
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 v ...  
J Main.java > ↗ Main  
1 class Area {  
2 // fields to calculate area  
3 int length;  
4 int breadth;  
5 // constructor to initialize values  
6 Area(int length, int breadth) {  
7 this.length = length;  
8 this.breadth = breadth;  
9 }  
10 // method to calculate area  
11 public void getArea() {  
12 int area = length * breadth;  
13 System.out.println("Area: " + area);  
14 }  
15 class Main {  
16 Run | Debug  
17 public static void main(String[] args) {  
18 }  
19 }  
PROBLEMS ② 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  
Area: 30  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [ ]
```

The status bar at the bottom indicates "Ln 28, Col 2 Spaces: 2 Off-5 CRTL ⌘ Java ⌘ Prettier".

### ● Java OOP (III)

### ✓ Java Nested and Inner Class

The screenshot shows the Visual Studio Code interface with the title bar "Review PBO\_11S23043". The left sidebar shows a project tree with files like Main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, PassOrFail.java, and Assignment. The main editor window displays the following Java code:

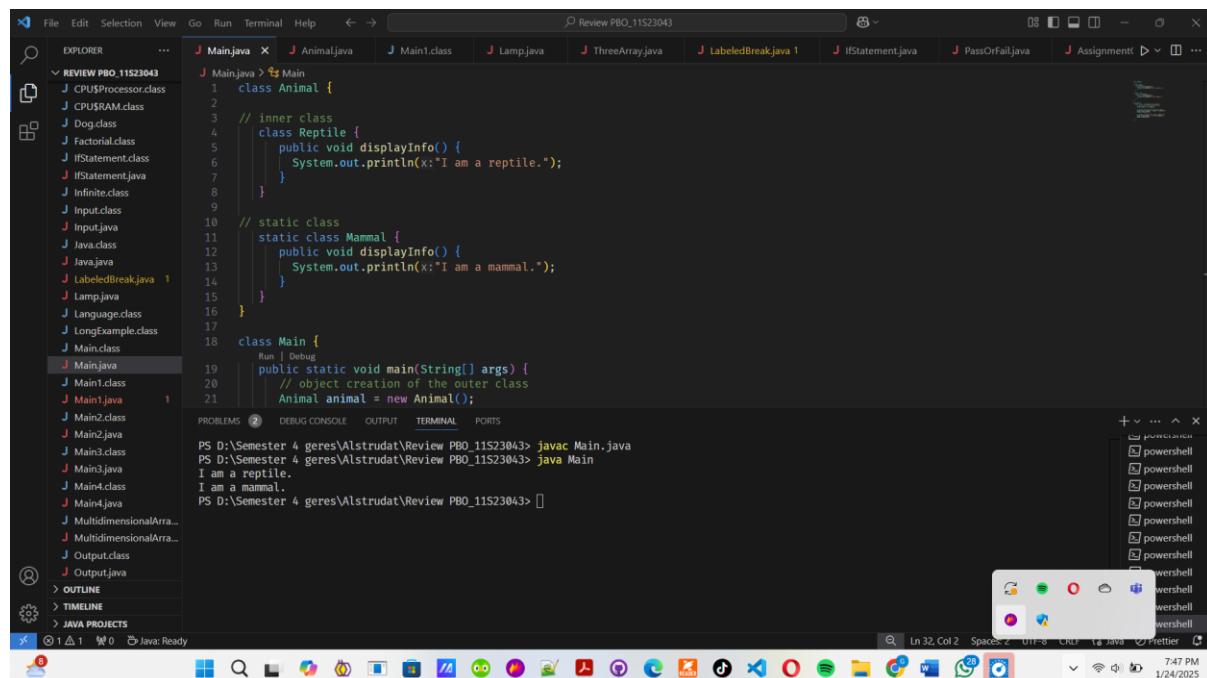
```
J Main.java x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D v ...  
J Main.java > ↗ CPU  
1 class CPU {  
2 double price;  
3 // nested class  
4 class Processor{  
5 // members of nested class  
6 double cores;  
7 String manufacturer;  
8  
9 double getCache(){  
10 return 4.3;  
11 }  
12 }  
13 }  
14 // nested protected class  
15 protected class RAM{  
16 // members of protected nested class  
17 double memory;  
18 String manufacturer;  
19  
20 double getClockSpeed(){  
21 return 5.5;  
22 }  
PROBLEMS ② 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  
Processor Cache = 4.3  
Ram Clock speed = 5.5  
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> [ ]
```

The status bar at the bottom indicates "Ln 15, Col 30 Spaces: 2 Off-5 CRTL ⌘ Java ⌘ Prettier".

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java Nested Static Class



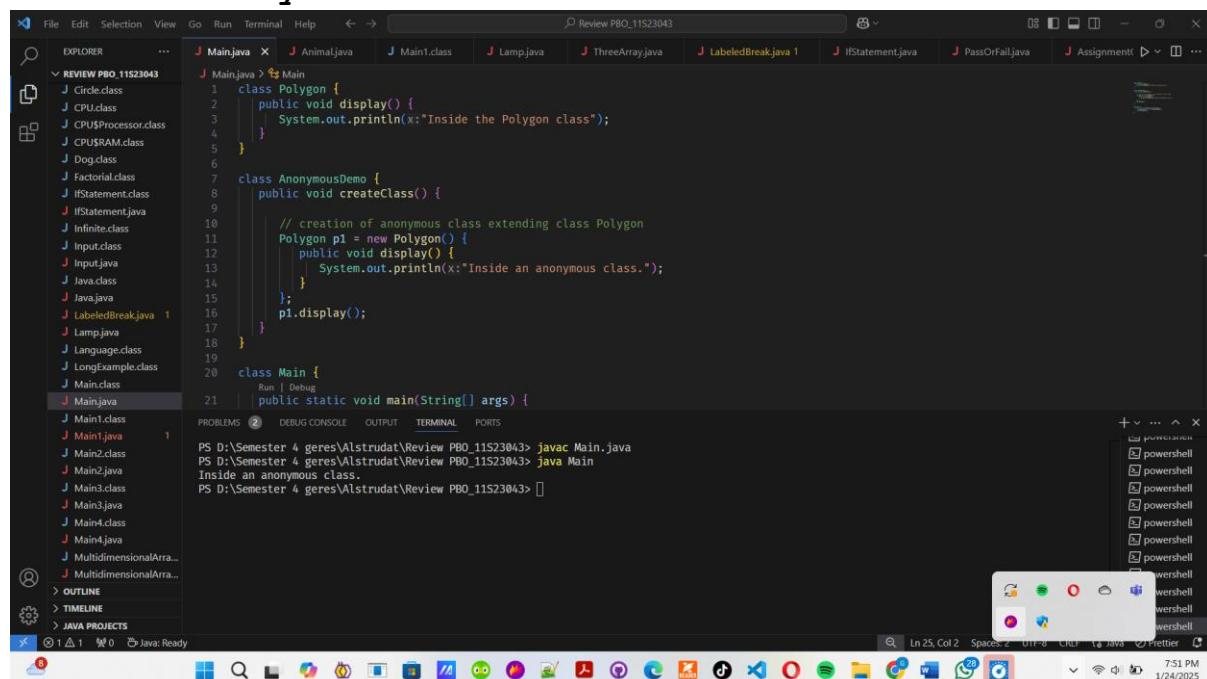
The screenshot shows the Eclipse IDE interface with the following code in the Main.java file:

```
1 class Animal {
2     // inner class
3     class Reptile {
4         public void displayInfo() {
5             System.out.println("I am a reptile.");
6         }
7     }
8 }
9
10 // static class
11 static class Mammal {
12     public void displayInfo() {
13         System.out.println("I am a mammal.");
14     }
15 }
16
17 class Main {
18     Run | Debug
19     public static void main(String[] args) {
20         // object creation of the outer class
21         Animal animal = new Animal();
22     }
23 }
```

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 Eclipse IDE interface with the following code in the Main.java file:

```
1 class Polygon {
2     public void display() {
3         System.out.println("Inside the Polygon class");
4     }
5 }
6
7 class AnonymousDemo {
8     public void createClass() {
9         // creation of anonymous class extending class Polygon
10        Polygon p1 = new Polygon() {
11            public void display() {
12                System.out.println("Inside an anonymous class.");
13            }
14        };
15        p1.display();
16    }
17 }
18
19 class Main {
20     Run | Debug
21     public static void main(String[] args) {
22     }
23 }
```

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 VS Code interface with the following details:

- File Explorer:** Shows files in the project: Circle.class, CPU.class, CPUSProcessor.class, CPUSRAM.class, Database.class, Dog.class, Factorial.class, IfStatement.class, IfStatement.java, Infinite.class, Input.class, Input.java, Java.class, Java.java, LabeledBreak.java, Lamp.java, Language.class, LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Main5.java, MultidimensionalArra...
- Editor:** Displays the `Main.java` file content:

```
1  class Database {  
2      private static Database dbObject;  
3  
4      private Database() {  
5          //  
6      }  
7  
8      public static Database getInstance() {  
9          // create object if it's not already created  
10         if(dbObject == null) {  
11             dbObject = new Database();  
12         }  
13  
14         // returns the singleton object  
15         return dbObject;  
16     }  
17  
18     public void getConnection() {  
19         System.out.println("You are now connected to the database.");  
20     }  
21 }
```
- Terminal:** Shows 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  
You are now connected to the database.  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom:** Shows the Windows taskbar with various pinned icons.

### ✓ Java enums

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files in the project: Circle.class, CPU.class, CPUSProcessor.class, CPUSRAM.class, Database.class, Dog.class, Factorial.class, IfStatement.class, IfStatement.java, Infinite.class, Input.class, Input.java, Java.class, Java.java, LabeledBreak.java, Lamp.java, Language.class, LongExample.class, Main.class, Main.java, Main1.class, Main1.java, Main2.class, Main2.java, Main3.class, Main3.java, Main4.class, Main4.java, Main5.java, MultidimensionalArra...
- Editor:** Displays the `Main.java` file content:

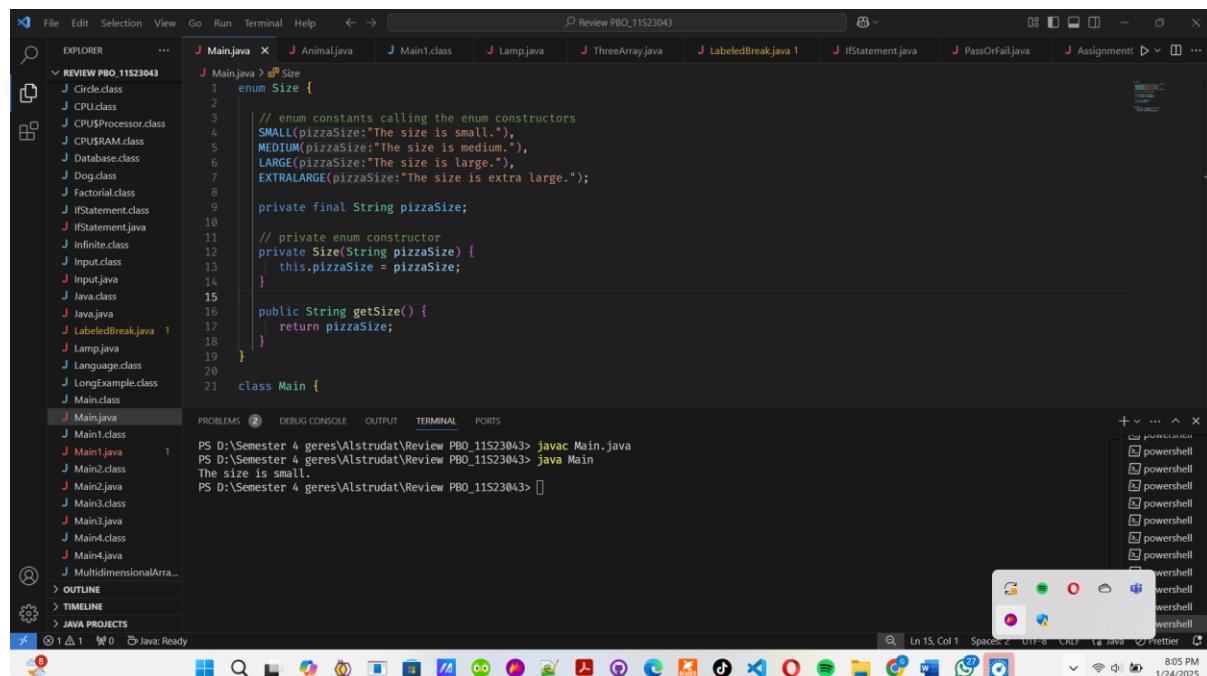
```
1  enum Size {  
2      SMALL, MEDIUM, LARGE, EXTRALARGE  
3  }  
4  
5  class Main {  
6      public static void main(String[] args) {  
7          System.out.println(Size.SMALL);  
8          System.out.println(Size.MEDIUM);  
9      }  
10 }
```
- Terminal:** Shows command-line output:

```
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main  
SMALL  
MEDIUM  
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
```
- Bottom:** Shows the Windows taskbar with various pinned icons.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java enum Constructor



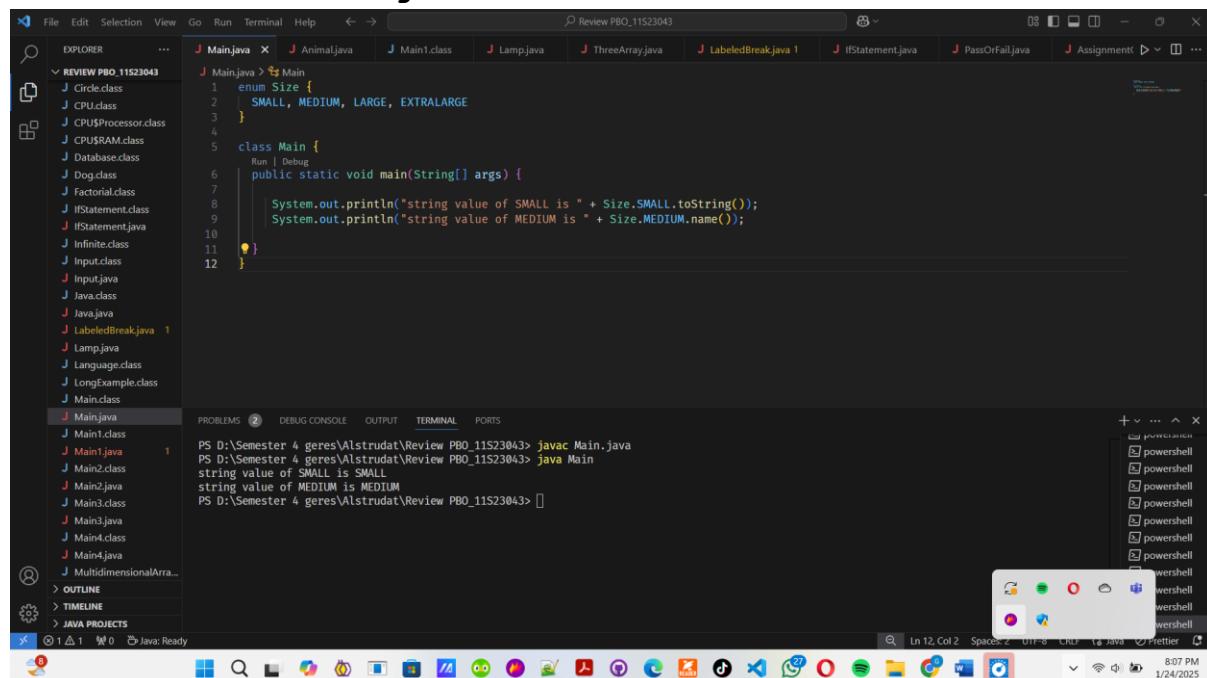
```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER ... J Main.java x J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J PassOrFail.java J Assignment D ...
J Main.java > Size
1 enum Size {
2     // enum constants calling 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 {
```

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>
```

Ln 15, Col 1 Spaces: 2 0ff-5 CWD Java prettier

### ✓ Java enum Strings



```
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER ... 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 ...
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
8         System.out.println("string value of SMALL is " + Size.SMALL.toString());
9         System.out.println("string value of MEDIUM is " + Size.MEDIUM.name());
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>
```

Ln 12, Col 2 Spaces: 2 0ff-5 CWD Java prettier

✓ **Java Package**

```

File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER J DogJava J Main.class J Main.java J AnimalJava J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java
J Dog.java J Factorial.class J IfStatement.class 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 MultidimensionalArr... J MultidimensionalArr... J Output.class J Output.java J PassOrFail.class J PassOrFail.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
[3, 2, 1]
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>

```

● **Java Exception Handling**

```

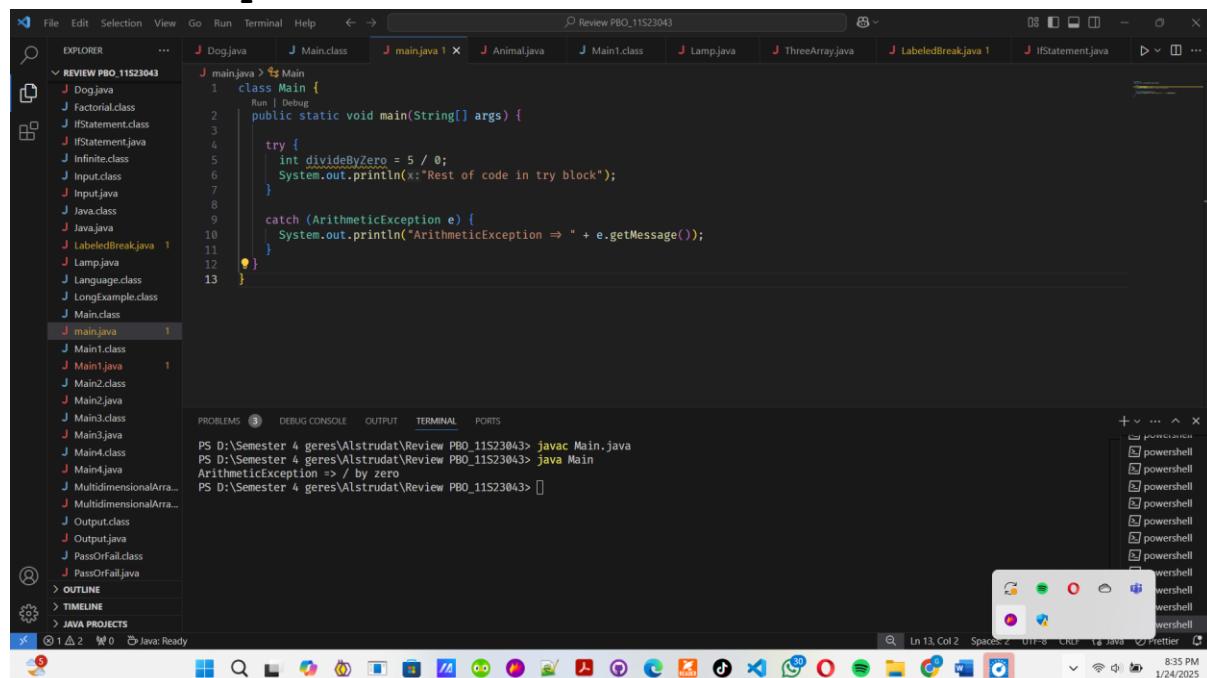
File Edit Selection View Go Run Terminal Help <- > Review PBO_11S23043
EXPLORER J DogJava J Main.class J main.java 1 X J AnimalJava J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java
J Dog.java J Factorial.class J IfStatement.class J Infinite.class J Input.class J Input.java 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 1 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 MultidimensionalArr... J MultidimensionalArr... J Output.class J Output.java J PassOrFail.class J PassOrFail.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
ArithmaticException => / by zero
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>

```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java try...catch



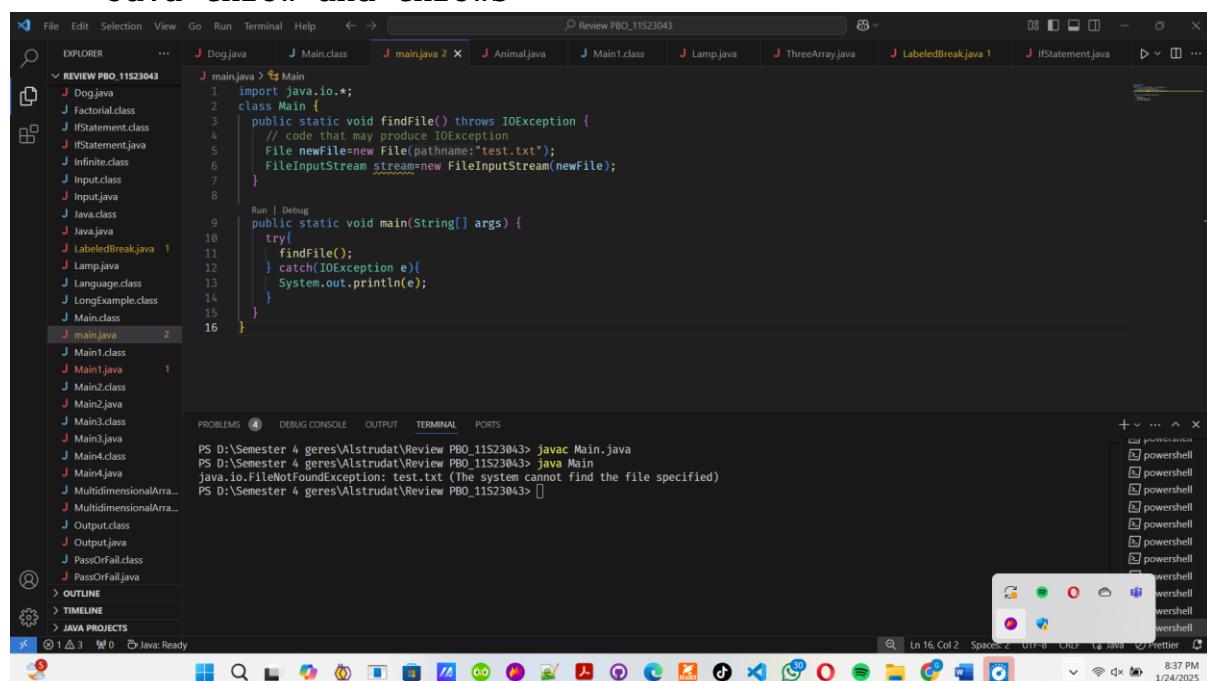
```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER ... J Dog.java J Main.class J main.java 1 ✘ J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java
J Dog.java
J Factorial.class
J IfStatement.class
J IfStatement.java
J Infinite.class
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 1
J Main1.class
J Main.java 1
J Main2.class
J Main2.java
J Main3.class
J Main3.java
J Main4.class
J Main4.java
J MultidimensionalArr...
J MultidimensionalArr...
J Output.class
J Output.java
J PassOrFail.class
J PassOrFail.java
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
ArithmaticException => / by zero
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
Ln 13, Col 2 Spaces: 2 0ff-5 CRLT Java ⏎ PRETTIER
Java: Ready
```

The screenshot shows the Visual Studio Code interface with the title bar "Review PBO\_11S23043". The Explorer sidebar lists various Java files. The main editor window contains the following Java code:

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

The terminal below shows the command-line output of running the Java compiler and the resulting application. An error message "ArithmaticException => / by zero" is displayed.

### ✓ Java throw and throws



```
File Edit Selection View Go Run Terminal Help ⏎ → Review PBO_11S23043
EXPLORER ... J Dog.java J Main.class J main.java 2 ✘ J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java
J Dog.java
J Factorial.class
J IfStatement.class
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 2
J Main1.class
J Main.java 1
J Main2.class
J Main2.java
J Main3.class
J Main3.java
J Main4.class
J Main4.java
J MultidimensionalArr...
J MultidimensionalArr...
J Output.class
J Output.java
J PassOrFail.class
J PassOrFail.java
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
java.io.FileNotFoundException: test.txt (The system cannot find the file specified)
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043>
Ln 16, Col 2 Spaces: 2 0ff-5 CRLT Java ⏎ PRETTIER
Java: Ready
```

The screenshot shows the Visual Studio Code interface with the title bar "Review PBO\_11S23043". The Explorer sidebar lists various Java files. The main editor window contains the following Java code:

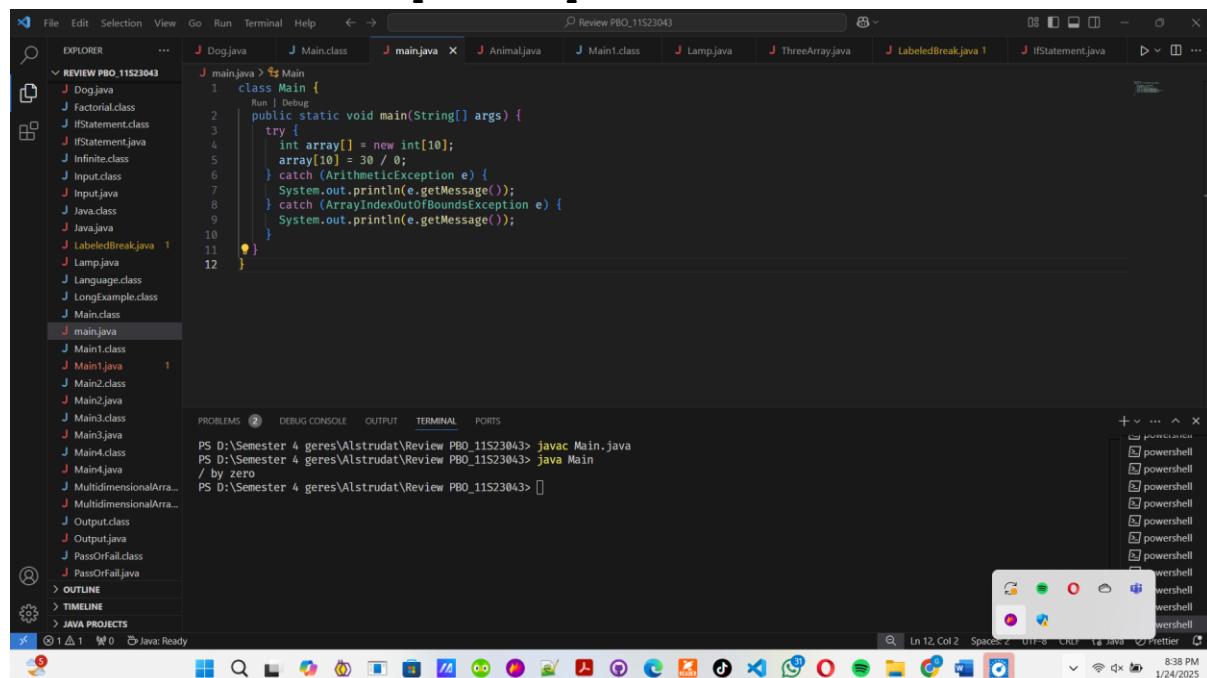
```
import java.io.*;
class Main {
    public static void findFile() throws IOException {
        // code that may produce IOException
        File newFile=new File(pathname:"test.txt");
        FileInputStream _stream=new FileInputStream(newFile);
    }
    public static void main(String[] args) {
        try{
            findFile();
        } catch(IOException e){
            System.out.println(e);
        }
    }
}
```

The terminal below shows the command-line output of running the Java compiler and the resulting application. An error message "java.io.FileNotFoundException: test.txt (The system cannot find the file specified)" is displayed.

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java catch Multiple Exceptions

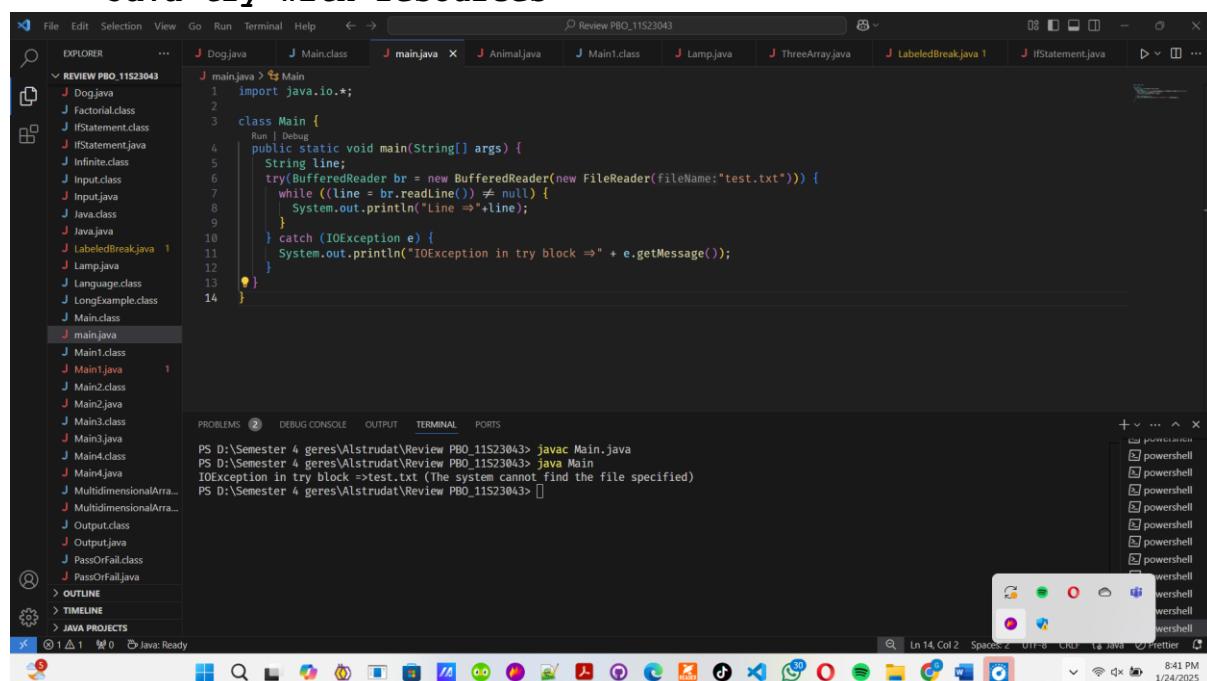


```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Review PBO_11S23043
REVIEW PBO_11S23043
Dog.java Main.class main.java Animal.java Main1.class Lamp.java ThreeArray.java LabeledBreak.java 1 IfStatement.java
J Dog.java J Main.class J main.java X J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java
J Factorial.class J IfStatement.class 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 Main.java 1 J Main2.class J Main3.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
@ 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
/ by zero
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
12:00 PM 1/24/2025
```

The code in the main.java file is:

```
class Main {
    public static void main(String[] args) {
        try {
            int array[] = new int[10];
            array[10] = 30 / 0;
        } catch (ArithmaticException e) {
            System.out.println(e.getMessage());
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

### ✓ Java try-with-resources



```
File Edit Selection View Go Run Terminal Help ⏮ ⏯ Review PBO_11S23043
REVIEW PBO_11S23043
Dog.java Main.class main.java Animal.java Main1.class Lamp.java ThreeArray.java LabeledBreak.java 1 IfStatement.java
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 Factorial.class J IfStatement.class 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 Main.java 1 J Main2.class J Main3.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
@ 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
IOException in try block =>test.txt (The system cannot find the file specified)
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>
8:41 PM 1/24/2025
```

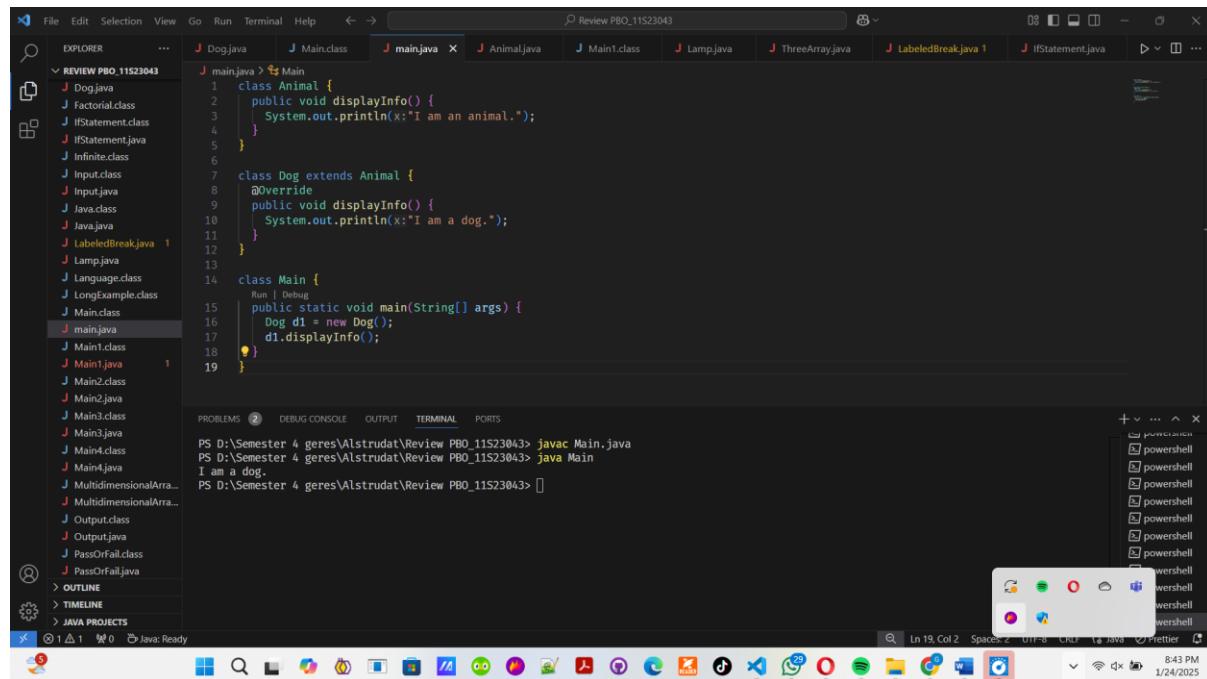
The code in the main.java file is:

```
import java.io.*;
class Main {
    public static void main(String[] args) {
        String line;
        try(BufferedReader br = new BufferedReader(new FileReader(fileName:"test.txt"))){
            while ((line = br.readLine()) != null) {
                System.out.println("Line =>" + line);
            }
        } catch (IOException e) {
            System.out.println("IOException in try block =>" + e.getMessage());
        }
    }
}
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java Annotations



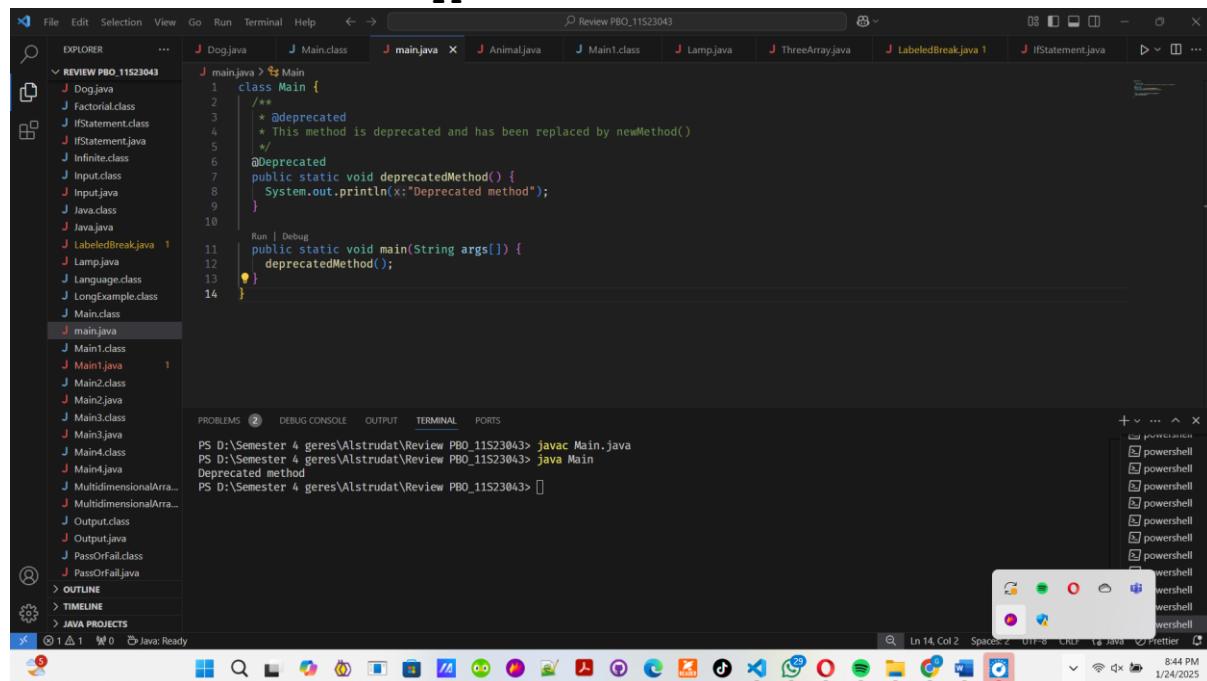
A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a project tree with files like Dog.java, Main.class, main.java, Animal.java, Main1.class, Lamp.java, ThreeArray.java, LabeledBreak.java, IfStatement.java, and Main.java. The main editor tab is on main.java, displaying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

```
J 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 tab 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 Main.java. The main editor tab is on main.java, displaying the following code:

```
J 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 }
```

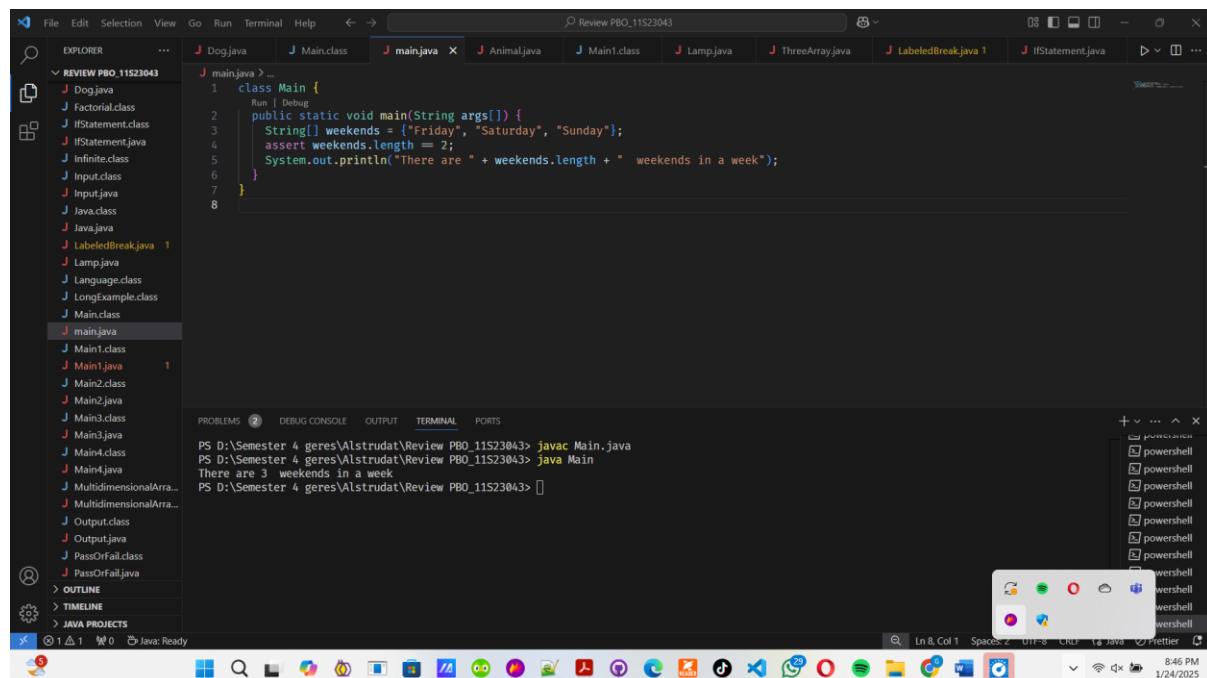
The terminal tab at the bottom shows the output of running the code:

```
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



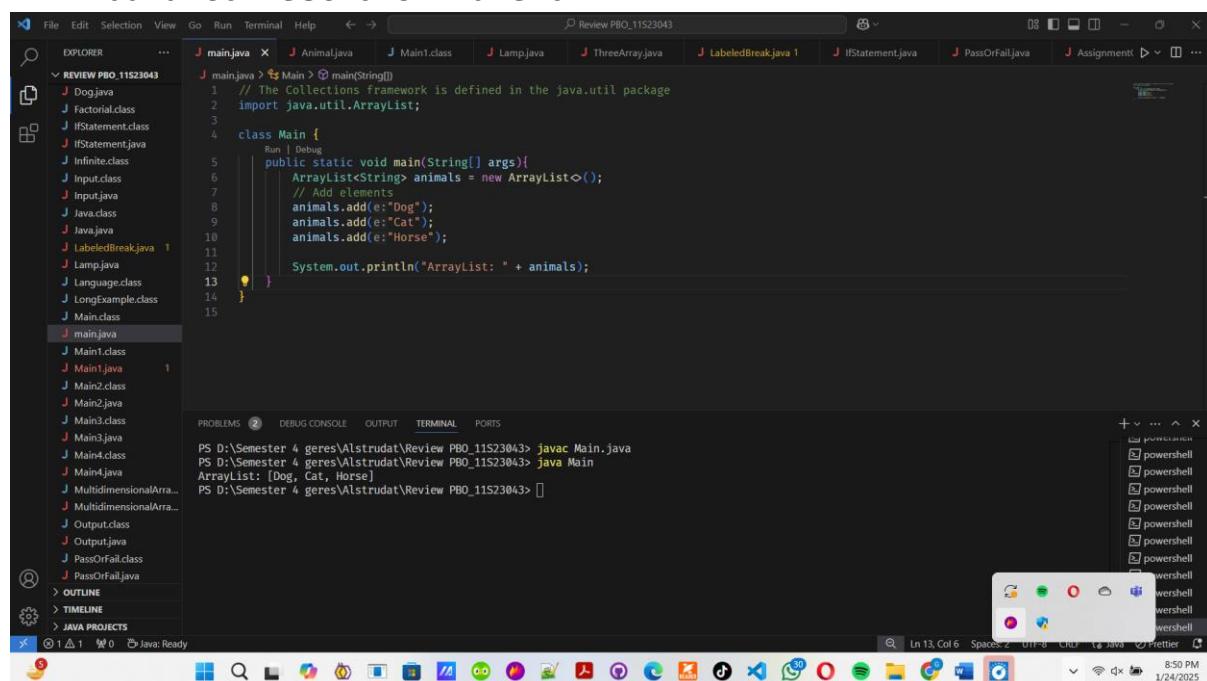
```
File Edit Selection View Go Run Terminal Help ↻ → Review PBO_11S23043
EXPLORER ... J Dog.java J Main.class J main.java X J Animal.java J Main1.class J Lamp.java J ThreeArray.java J LabeledBreak.java 1 J IfStatement.java J Infinite.class J Input.class J Input.java J Java.class J Java.java J LabeledBreak.java 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 > OUTLINE > TIMELINE > 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 Off: 3 CRTL ⌘ ⌘ Java ⌘ Fetter ⌘
8:45 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 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 > OUTLINE > TIMELINE > 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

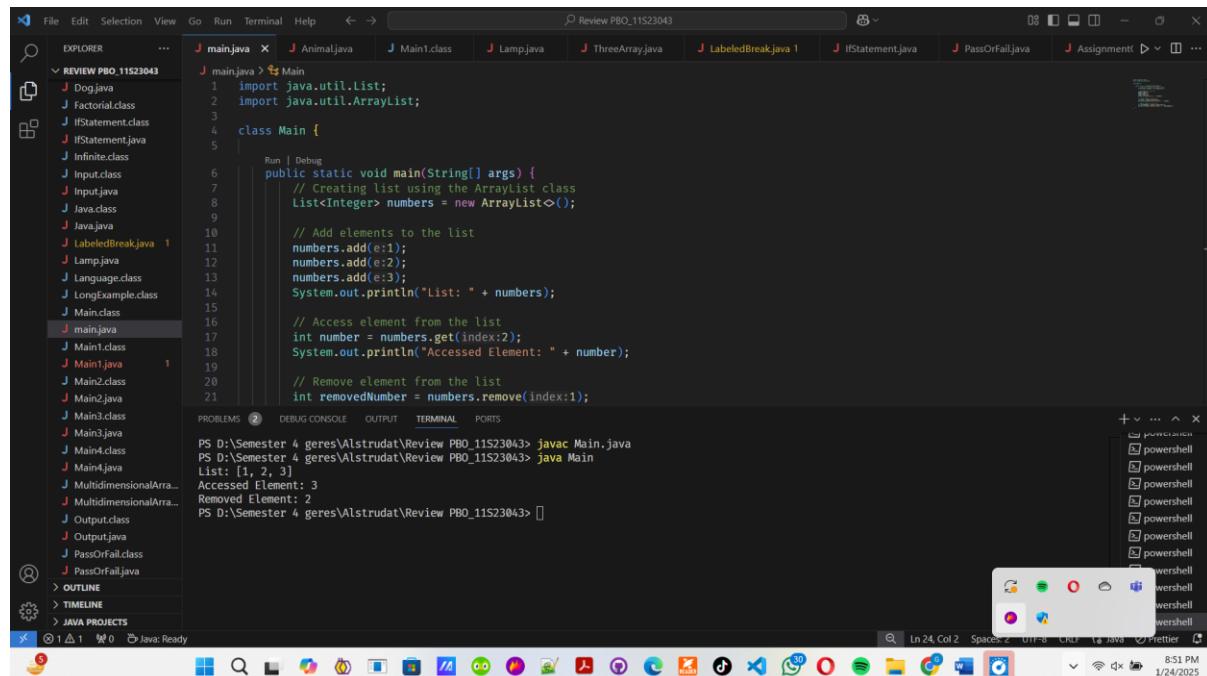
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> javac Main.java
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043> java Main
ArrayList: [Dog, Cat, Horse]
PS D:\Semester 4\geres\Alstrudat\Review PBO_11S23043>

Ln 13, Col 6 Spaces: 2 Off: 3 CRTL ⌘ ⌘ Java ⌘ Fetter ⌘
8:50 PM 1/24/2025
```

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java List



The screenshot shows a Java application named Main. The code uses an ArrayList to store integers and prints them to the console. The terminal output shows the list [1, 2, 3] and the accessed element 2.

```
import java.util.List;
import java.util.ArrayList;

public static void main(String[] args) {
    // Creating list using the ArrayList class
    List<Integer> numbers = new ArrayList<>();

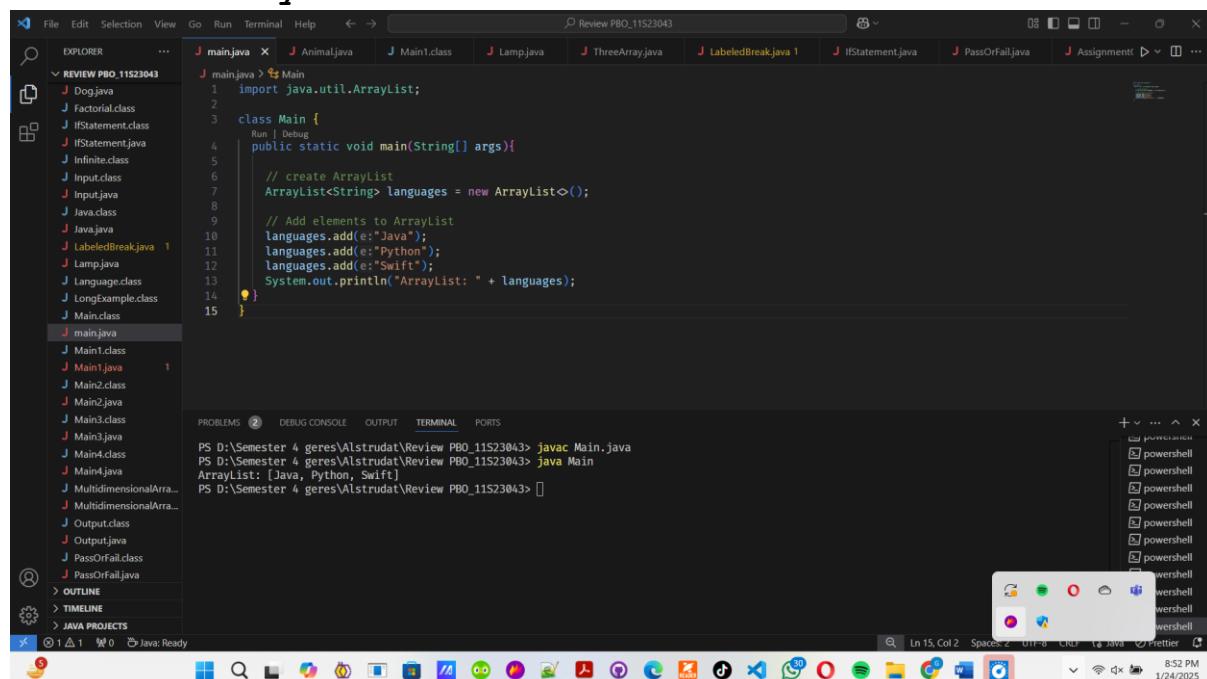
    // Add elements to the list
    numbers.add(1);
    numbers.add(2);
    numbers.add(3);
    System.out.println("List: " + numbers);

    // Access element from the list
    int number = numbers.get(index:2);
    System.out.println("Accessed Element: " + number);

    // Remove element from the list
    int removedNumber = numbers.remove(index:1);
}
```

PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> javac Main.java  
PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> java Main  
List: [1, 2, 3]  
Accessed Element: 2  
Removed Element: 2  
PS D:\Semester 4 geres\Alstrudat\Review PBO\_11S23043> []

### ✓ Java ArrayList



The screenshot shows a Java application named Main. The code creates an ArrayList of strings containing Java, Python, and Swift, and prints it to the console. The terminal output shows the list [Java, Python, Swift].

```
import java.util.ArrayList;

public static void main(String[] args){
    // create ArrayList
    ArrayList<String> languages = new ArrayList<>();

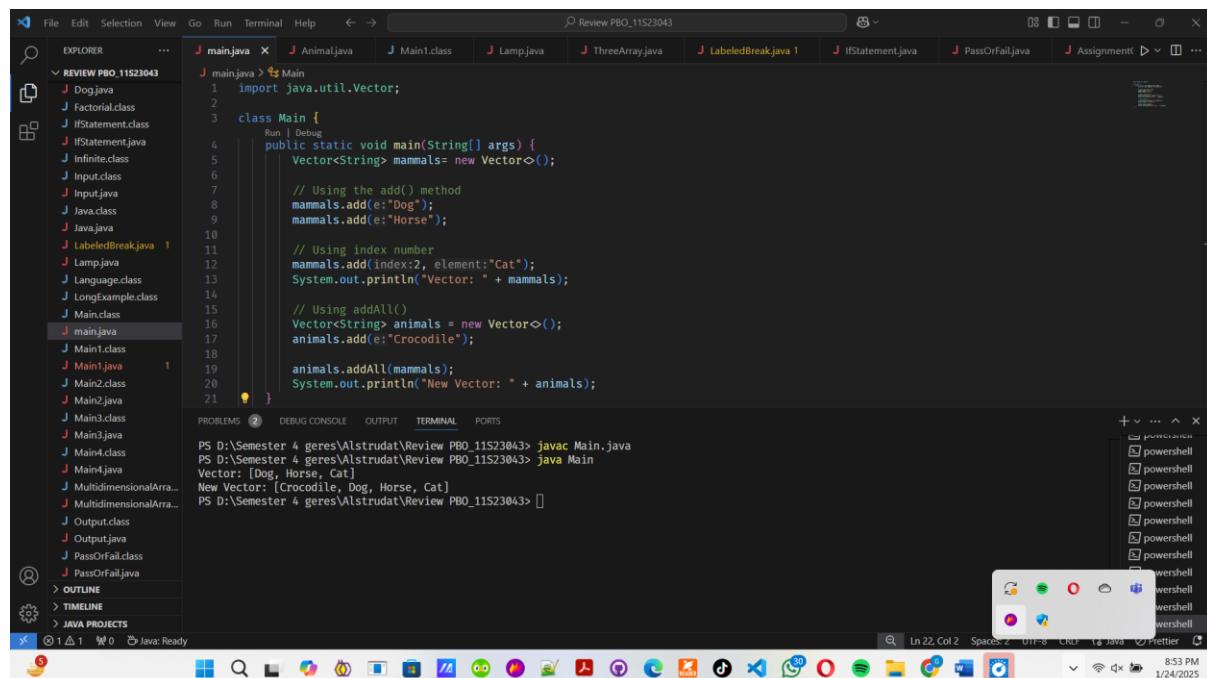
    // Add elements to ArrayList
    languages.add(e:"Java");
    languages.add(e:"Python");
    languages.add(e:"Swift");
    System.out.println("ArrayList: " + languages);
}
```

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> []

# 11S2215 - Algorithms and Data Structures

## Laporan Praktikum

### ✓ Java Vector



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

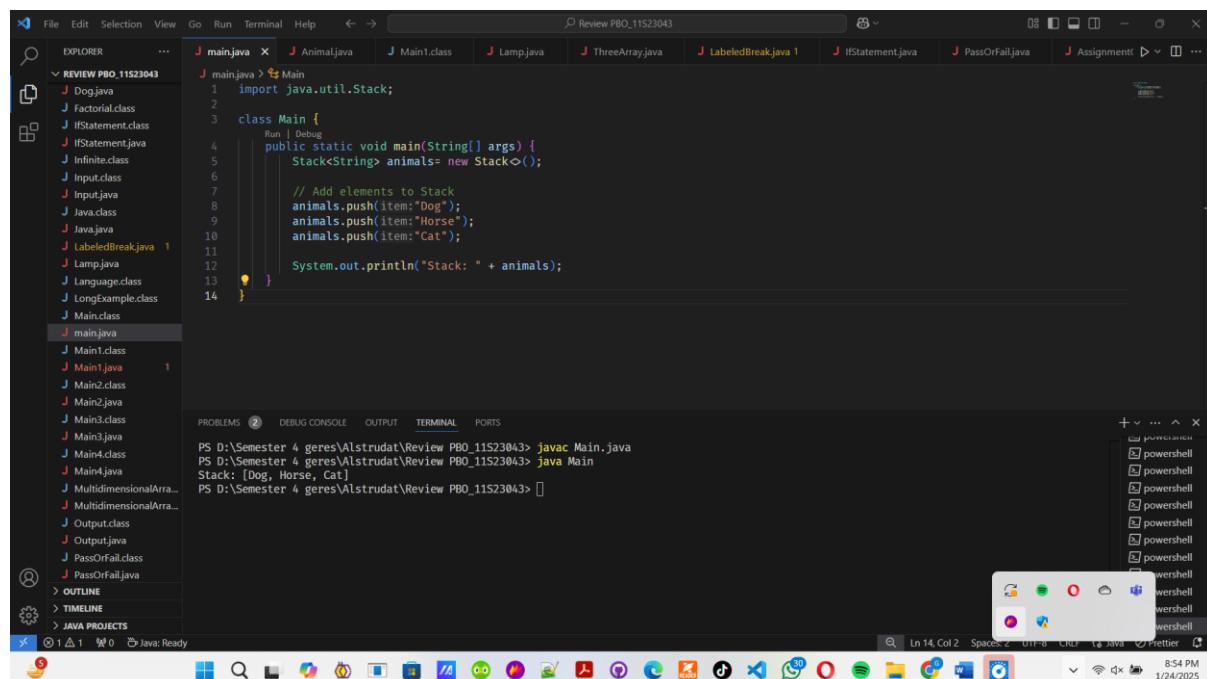
- File Explorer:** Shows a project named "REVIEW PBO\_11S23043" containing various Java files like Dog.java, IfStatement.class, Input.class, etc.
- Code Editor:** Displays the "main.java" file with the following code:

```
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.java

J main.java > Main
1 import java.util.Vector;
2
3 class Main {
4     Run | Debug
5     public static void main(String[] args) {
6         Vector<String> mammals= new Vector<>();
7
8         // Using the add() method
9         mammals.add("Dog");
10        mammals.add("Horse");
11
12        // Using index number
13        mammals.add(index:2, element:"Cat");
14        System.out.println("Vector: " + mammals);
15
16        // Using addAll()
17        Vector<String> animals = new Vector<>();
18        animals.add("Crocodile");
19
20        animals.addAll(mammals);
21        System.out.println("New Vector: " + animals);
22    }
23}
```
- Terminal:** Shows the command line 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
Vector: [Dog, Horse, Cat]
New Vector: [Crocodile, Dog, Horse, Cat]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> []
```
- Taskbar:** Shows the Windows taskbar with various pinned icons including File Explorer, Edge, and File History.

### ✓ 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, IfStatement.class, Input.class, etc.
- Code Editor:** Displays the "main.java" file with the following code:

```
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.java

J main.java > Main
1 import java.util.Stack;
2
3 class Main {
4     Run | Debug
5     public static void main(String[] args) {
6         Stack<String> animals= new Stack<>();
7
8         // Add elements to Stack
9         animals.push(item:"Dog");
10        animals.push(item:"Horse");
11        animals.push(item:"Cat");
12
13    }
14}
```
- Terminal:** Shows the command line 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
Stack: [Dog, Horse, Cat]
PS D:\Semester 4 geres\Alstrudat\Review PBO_11S23043> []
```
- Taskbar:** Shows the Windows taskbar with various pinned icons including File Explorer, Edge, and File History.

# **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 Explorer:** Shows a project named "IF52043-ALSTRUDAT-P1" containing files: Matrix.java, MatrixApp.java, and MatrixApp.java (1).
- Code Editor:** Displays the content of Matrix.java. The code defines a Matrix class with methods for setting elements, getting elements, displaying the matrix, and adding another matrix.
- Bottom Status Bar:** Shows "Ln 42 Col 8 Java Ready".

```
File Edit Selection View Go Run Terminal Help < > if52043-alstrudat-p1

EXPLORER ... J Matrix.java J MatrixApp.java J MatrixApp.java D:\...\Vs23032-alstrudat-p1

J Matrix.java > Matrix > add(Matrix)
1 public class Matrix {
2     private int [][] elemen;
3     private int baris;
4     private int kolom;
5     private String name;
6
7     public Matrix (int baris, int kolom, String name){
8         this.baris = baris;
9         this.kolom = kolom;
10        this.elemen = new int [baris][kolom];
11        this.name = name;
12    }
13
14    public void settlemen (int baris1, int kolom1, int nilai){
15        elemen[baris1][kolom1] = nilai;
16    }
17
18    public int getElement (int baris1, int kolom1){
19        return elemen[baris1][kolom1];
20    }
21
22    public void display () {
23        System.out.println("Matrix " + name + ":");
24        for (int i = 0; i < baris; i++){
25            for (int j = 0; j < kolom; j++){
26                System.out.print(elemen[i][j] + " ");
27            }
28            System.out.println();
29        }
30    }
31
32    public boolean add(Matrix B) {
33        if (this.baris != B.baris || this.kolom != B.kolom) {
34            System.out.println("Jumlah baris dan kolom untuk kedua matrix harus sama");
35            return false;
36        }
37
38        for (int i = 0; i < this.baris; i++) {
39            for (int j = 0; j < this.kolom; j++) {
40                this.settlemen(i, j, this.getElement(i, j) + B.getElement(i, j));
41            }
42        }
43        return true;
44    }
45 }

Ln 42 Col 8 Java Ready
```

## MatrixApp.java



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

- File Explorer:** Shows the project structure for "fis23043-alstrudat-p1".
- Editor:** Displays the content of `MatrixApp.java`. The code implements methods for addition, subtraction, and multiplication of matrices.
- Output:** Shows the terminal output area.
- Bottom Bar:** Includes icons for file operations like Open, Save, and Close, as well as other standard OS X-style controls.

```
File Edit Selection View Go Run Terminal Help < > ffs23043-alstrudat-p1
EXPLORER J Matrix.java J MatrixApp.java J MatrixApp.java D:\Vb\J3052-alstrudat-p1
output MatrixApp.java
Matrix.java MatrixApp.java
MatrixApp.java
public class MatrixApp {
    public static void main(String [] args) {
        matrices.get(target1 - 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();

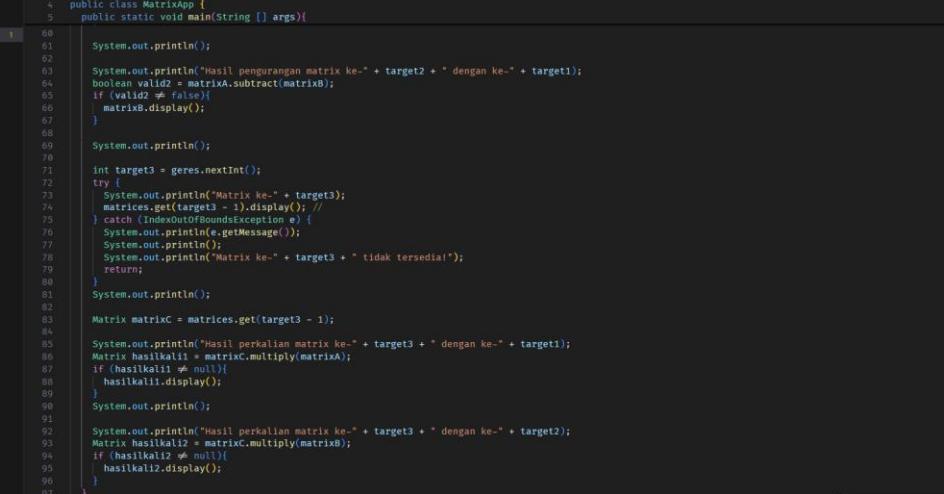
    int target3 = geres.nextInt();
    try {
        System.out.println("Matrix ke-" + target3);
        matrices.get(target3 - 1).display(); // 
    } catch (IndexOutOfBoundsException e) {
        System.out.println(e.getMessage());
        System.out.println();
        System.out.println("Matrix ke-" + target3 + " tidak tersedia!");
        return;
    }
    System.out.println();

    Matrix matrixC = matrices.get(target3 - 1);

    System.out.println("Hasil perkalian matrix ke-" + target3 + " dengan ke-" + target1);
    double hasil1 = matrixA.multiply(matrixC);
    double hasil2 = matrixC.multiply(matrixA);
}
```

# **11S2215 - Algorithms and Data Structures**

## Laporan Praktikum



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

- File Explorer:** Shows the project structure with files: Matrix.java, MatrixApp.java, and MatrixApp.java (diff). It also lists output and Matrix.java.
- Code Editor:** Displays Java code for matrix operations. The code includes methods for matrix subtraction, printing matrices, and matrix multiplication. It uses `MatrixX` objects and `System.out.println` for output.
- Terminal:** Shows the command `if523043-alstrudat-p1`.
- Status Bar:** Shows file paths like `if523043-alstrudat-p1`, line numbers (Ln 9 Col 34), and other status indicators.

Contoh program saat dijalankan:

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

- File Explorer:** Shows a project named "IF523043-ALSTRUDAT-PI" containing files: Matrix.java, MatrixApp.java, and MatrixApp.out.
- Code Editor:** Displays the Java code for `MatrixApp`. The code imports `java.util.ArrayList` and `java.util.Scanner`, defines a `MatrixApp` class with a `main` method, and uses a `Scanner` object to read input from `System.in`.
- Terminal:** Shows the command-line output of running the application:

```
PS D:\Semester 4 geres\Alstrudat\if523043-alstrudat-p1> java -d output MatrixApp.java Matrix.java
PS D:\Semester 4 geres\Alstrudat\if523043-alstrudat-p1> java -cp output MatrixApp
1
2
3
4

Buat matrix ke-1 dengan ukuran 2x2 dan judul A
1
2
3
4

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!
```
- Sidebar:** Includes sections for **OUTLINE**, **TIMELINE**, and **JAVA PROJECTS**.
- Bottom Bar:** Shows the taskbar with various icons.