JOBSHEET 13

Function Part 1

Name
Sherly Lutfi Azkiah Sulistyawati

NIM
2341720241

Class
1I

Department
Information Technology

Study Program
D4 Informatics Engineering

Labs Activity

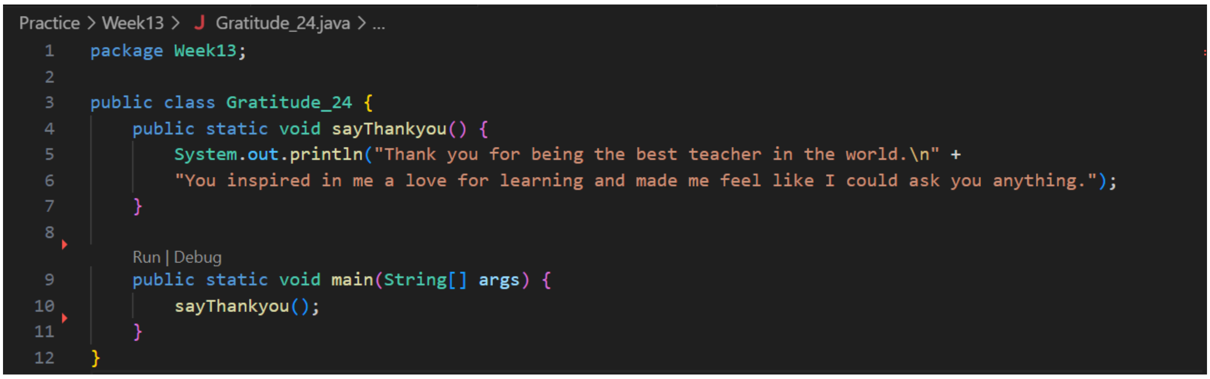
Question! (Experiment 1)

1. Does function with no parameter always have void datatype?
2. Is it possible for sentence “**Thank you for.....dst**” to be displayed, without using **sayThankyou()** function? Modify the program so that it displays the sentence without using function!
3. What are the benefits of using functions in a program?

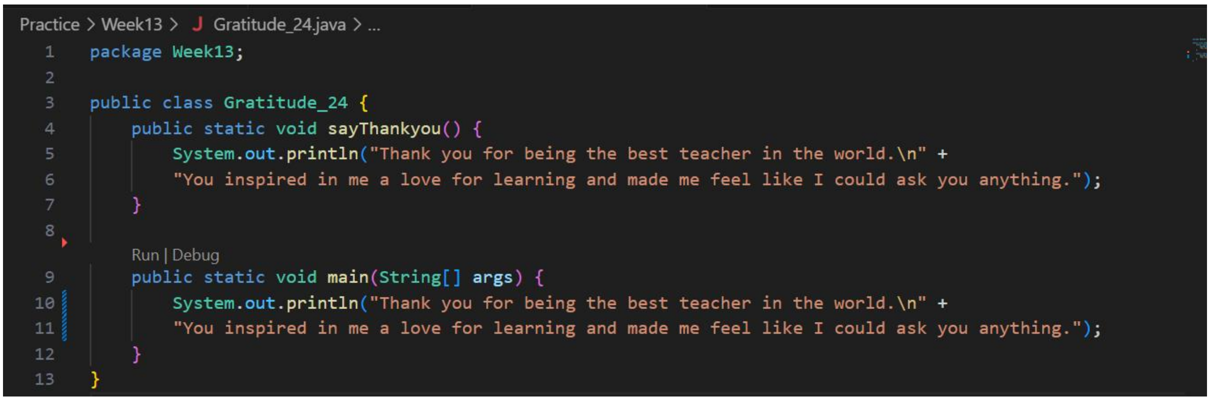
Answer!

a. No, because there is no relation between parameter and void datatype.

b. It is possible. We should copy the sentence into the main function.



```java
package Week13;

public class Gratitude_24 {
    public static void sayThankyou() {
        System.out.println("Thank you for being the best teacher in the world.\n" +
        "You inspired in me a love for learning and made me feel like I could ask you anything.");
    }

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



```java
package Week13;

public class Gratitude_24 {
    public static void sayThankyou() {
        System.out.println("Thank you for being the best teacher in the world.\n" +
        "You inspired in me a love for learning and made me feel like I could ask you anything.");
    }

    public static void main(String[] args) {
        System.out.println("Thank you for being the best teacher in the world.\n" +
        "You inspired in me a love for learning and made me feel like I could ask you anything.");
    }
}
```

c. Makes a code modular and more effective.

Question! (Experiment 2)

1. What is the use of a parameter in a function?
2. Is parameter similar to variable? Please explain!

3. In the Java programming language, is parameter only used for passing input data? What about output data?

Answer!

1. Parameters allow to input or pass values into a function and receive information from the calling code.
2. By the declaration are same, but the function are different. Variable are symbols for storing values in programs, while parameter are special variable used as input in function or method declarations. Variable can be used in various parts of a program, while parameter are a way to provide value to function.
3. Parameter in the Java programming language are used to pass data into a method. Also, the output value of the method can be returned using the return data type. So, parameters are not only for input, but also for output.

Question! (Experiment 3)

1. Explain when do we need to create a function that requires a return value?
2. Can a **System.out.println** statement be added inside a function with a return value? What is its impact?
3. Can a function without a return value be called inside the main function without being passed to a variable? Like in experiment 1? Explain!

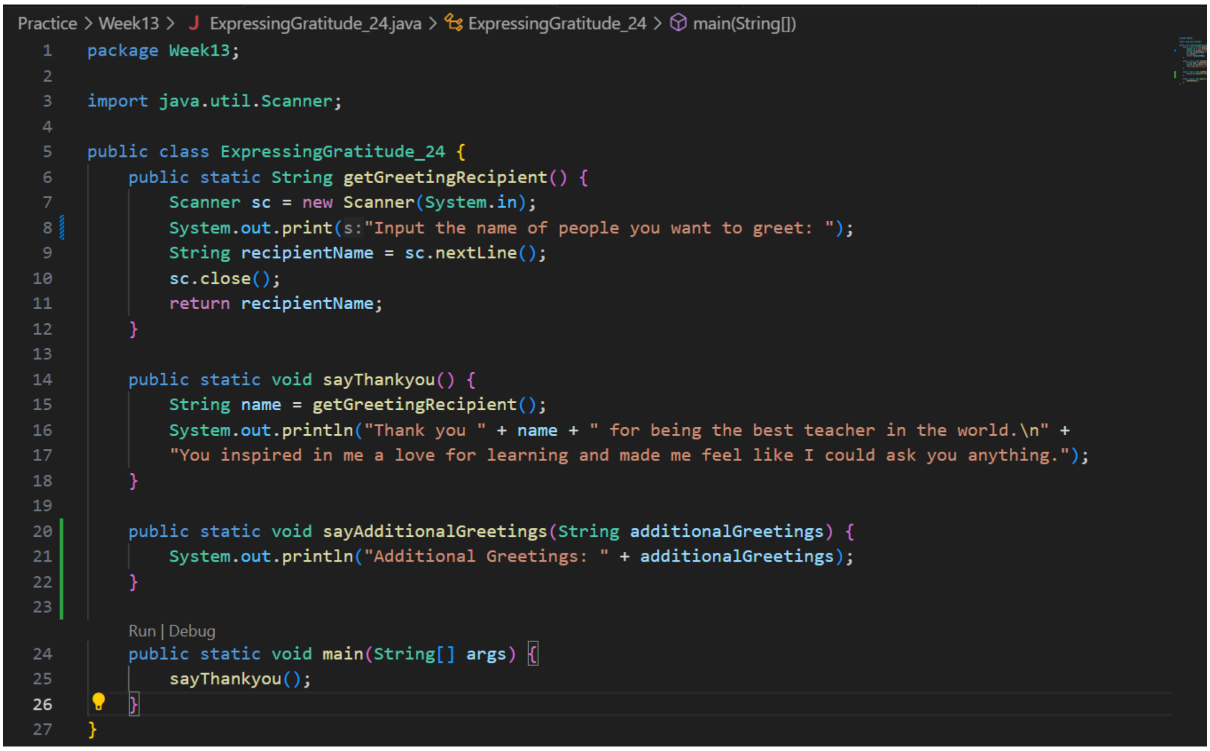
Answer!

1. Function is created to perform specific tasks or calculations, and they can either have a return value or not, depending on the requirements of the task.
2. Yes, we can add a System.out.println statement inside a function with a return value. However, keep in mind that this statement will only print to the console and not affect the function's return value. The effect is that the information printed to the console will be visible when the function is called, but the returned value will remain as expected.
3. Yes it can, a function without a return value (void functions) can be called directly in main() without assigning to a variable, like sayThankyou() in the previous examples. Since void functions do not return anything, there is no value to collect by assigning to a variable. So void functions can be invoked directly for their side effect (printing output) rather than capturing a return value.

Question! (Experiment 4)

1. Based on experiment 4, which function will execute first? Please explain!
2. Which is the correct way to write a function inside a class? Above the main function or below the main function? Please explain!
3. Modify the above program by adding the function sayAdditionalGreetings() with a String input parameter. The sayAdditionalGreetings() function contains additional remarks or greetings that you want to convey to the greeting recipient.

Answer!

1. In this program, the getGreetingRecipient() method will execute first. This is because it is called from inside the sayThankyou() method. So when sayThankyou() is invoked from main(), it starts by calling getgreetingRecipient() to get the name input first before it prints out the thank you message using that name
2. In Java, the correct way is to write functions above the main() method in a class. The order does not matter as long as functions are defined before they are called/used. By convention, we keep method definitions before the main() to enhance readability and maintainability.
3. 

Practice > Week13 > ExpressingGratitude_24.java > ExpressingGratitude_24 > main(String[])
Run | Debug

```java
package Week13;

import java.util.Scanner;

public class ExpressingGratitude_24 {
    public static String getGreetingRecipient() {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Input the name of people you want to greet: ");
        String recipientName = sc.nextLine();
        sc.close();
        return recipientName;
    }

    public static void sayThankyou() {
        String name = getGreetingRecipient();
        System.out.println("Thank you " + name + " for being the best teacher in the world.\n" +
        "You inspired in me a love for learning and made me feel like I could ask you anything.");
    }

    public static void sayAdditionalGreetings(String additionalGreetings) {
        System.out.println("Additional Greetings: " + additionalGreetings);
    }

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

Question! (Experiment 5)

1. Explain why the parameter in experiment 5 is written as **int... a**!

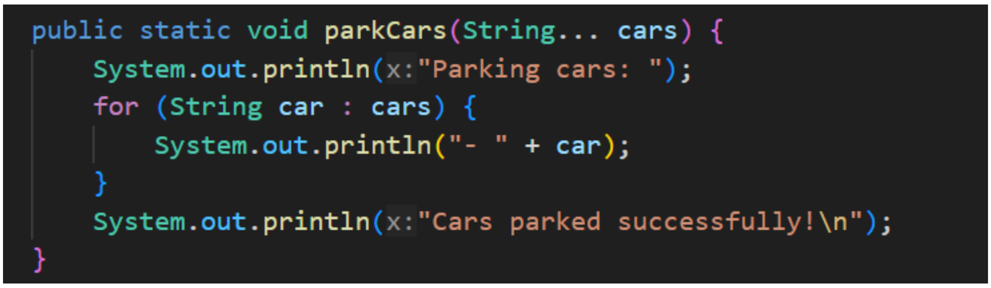
2. Mention the example of varargs in implementing code to solve real-world problems! (at least 3)

3. Can we use two different data types for varargs in one function? Provide an example!

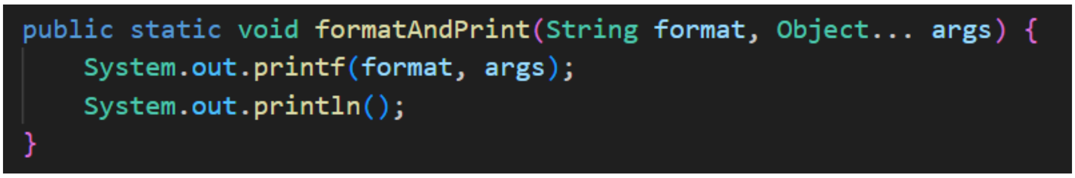
Answer!

1. The parameter in Experiment 5 is written as int... a to indicate the use of varargs (variable-length argument lists). The ellipsis (...) denotes that the method can accept a variable number of arguments of type int. This feature was introduced in Java 5, allowing a method to accept zero or more arguments of a specified type.

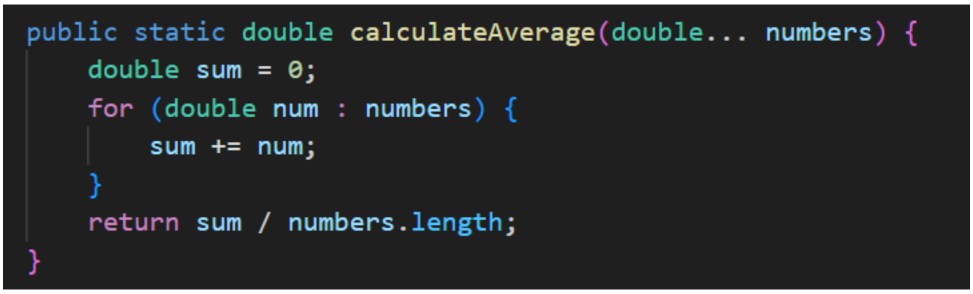
2.



```
public static void parkCars(String... cars) {
    System.out.println(x:"Parking cars: ");
    for (String car : cars) {
        System.out.println("- " + car);
    }
    System.out.println(x:"Cars parked successfully!\n");
}
```



```
public static void formatAndPrint(String format, Object... args) {
    System.out.printf(format, args);
    System.out.println();
}
```



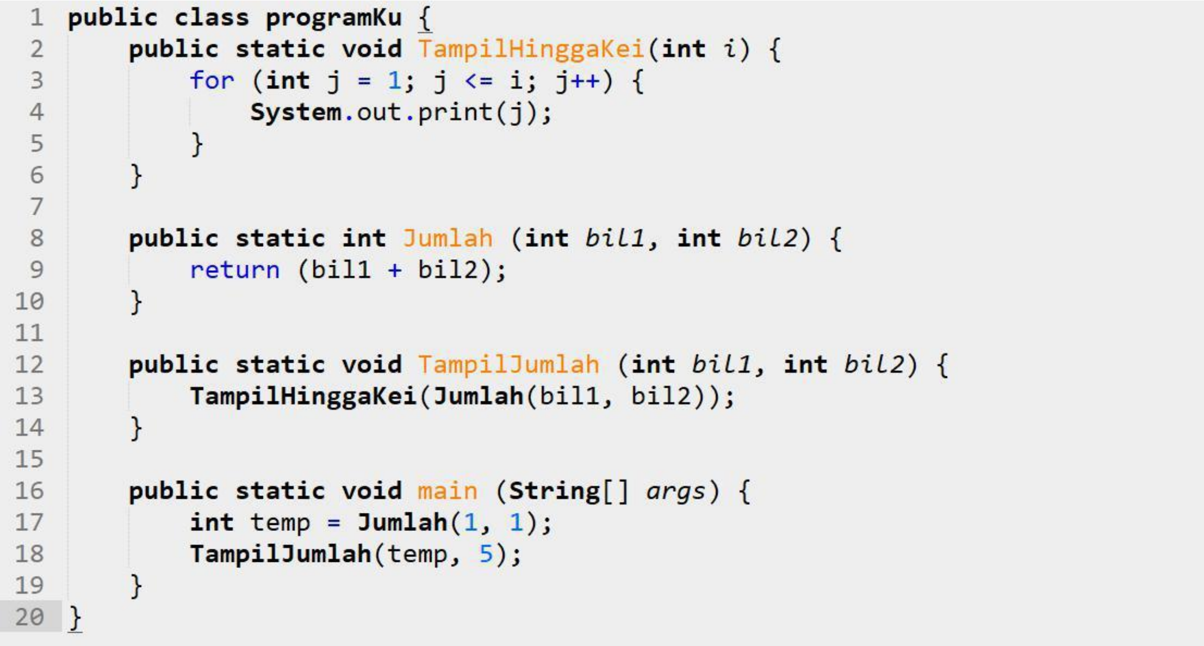
```
public static double calculateAverage(double... numbers) {
    double sum = 0;
    for (double num : numbers) {
        sum += num;
    }
    return sum / numbers.length;
}
```

3. No, a single varargs parameter in a function can only have one data type. However, you can achieve a similar effect by using an array or a collection that can hold elements of different types. Here's an example using an array of Object:

Question! (Experiment 6)

1. Explain the execution steps for experiment 6 above!

2. What is the output of the program below, then explain the steps of the program!



```java
public class programKu {
    public static void TampilHinggaKei(int i) {
        for (int j = 1; j <= i; j++) {
            System.out.print(j);
        }
    }

    public static int Jumlah (int bil1, int bil2) {
        return (bil1 + bil2);
    }

    public static void TampilJumlah (int bil1, int bil2) {
        TampilHinggaKei(Jumlah(bil1, bil2));
    }

    public static void main (String[] args) {
        int temp = Jumlah(1, 1);
        TampilJumlah(temp, 5);
    }
}
```

3. When do we need to create a function with and without parameters? When do we need to create a function with and without return value? Explain!

Answer!

1. Execution steps:

   - The main method is the starting point of execution.
   - It declares an integer variable temp and assigns the result of calling the Jumlah method with arguments 1 and 1.
   - It then calls the TampilJumlah method with arguments temp and 5.
   - The TampilJumlah method, in turn, calls the TampilHinggaKei method with the result of calling the Jumlah method as an argument.
   - Inside the TampilHinggaKei method, a for loop is executed to print numbers from 1 to the value passed as an argument.
   - The Jumlah method returns the sum of its two integer parameters.
   - The output is printed to the console.

2. The output of the program is "1 2 3 4 5 6". This is because:

   - The main method calculates the sum of 1 and 1, resulting in temp being assigned the value 2.
   - The TampilJumlah method calls TampilHinggaKei with the result of Jumlah(2, 5).
   - Inside TampilHinggaKei, a loop prints numbers from 1 to 7 (the result of 2 + 5).

3. **Functions with Parameters**

   - Use when you need to pass information to the function for it to operate on.
   - Parameters provide inputs to the function.
   - Example in the code: TampilHinggaKei(int i), Jumlah(int bil1, int bil2)

   Functions without Parameters

   - Use when the function doesn't require any external information to operate.
   - It performs its tasks based on internal logic.
   - Example in the code: main(String[] args)

   Functions with Return Value

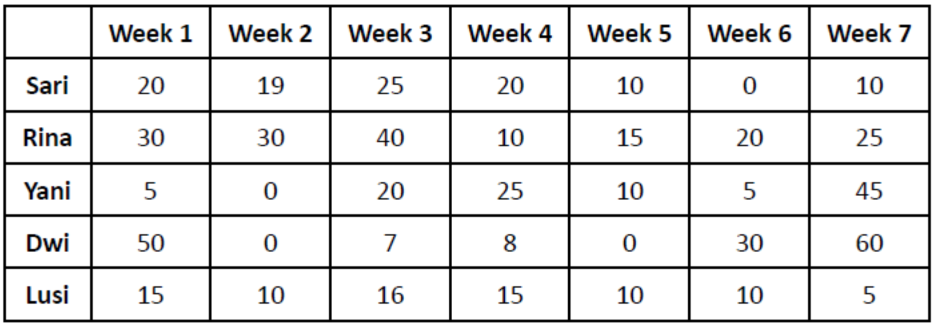
   - Use when the function needs to provide a result back to the caller.
   - It allows the function to produce an output that can be used elsewhere.
   - Example in the code: Jumlah(int bil1, int bil2)

   Functions without Return Value

   - Use when the function performs a task but doesn't need to provide any result.
   - It can have side effects or modify external state.
   - Example in the code: TampilHinggaKei(int i), TampilJumlah(int bil1, int bil2)

Assignment

1. Create a new class named CubeStudentIDNumber that has function to calculate area and volume!
2. Create a program to manage the weekly grades (there are 7 weeks) of 5 students. The data must be implemented using 2 dimensional array as follows:



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Sari	20	19	25	20	10	0	10
Rina	30	30	40	10	15	20	25
Yani	5	0	20	25	10	5	45
Dwi	50	0	7	8	0	30	60
Lusi	15	10	16	15	10	10	5

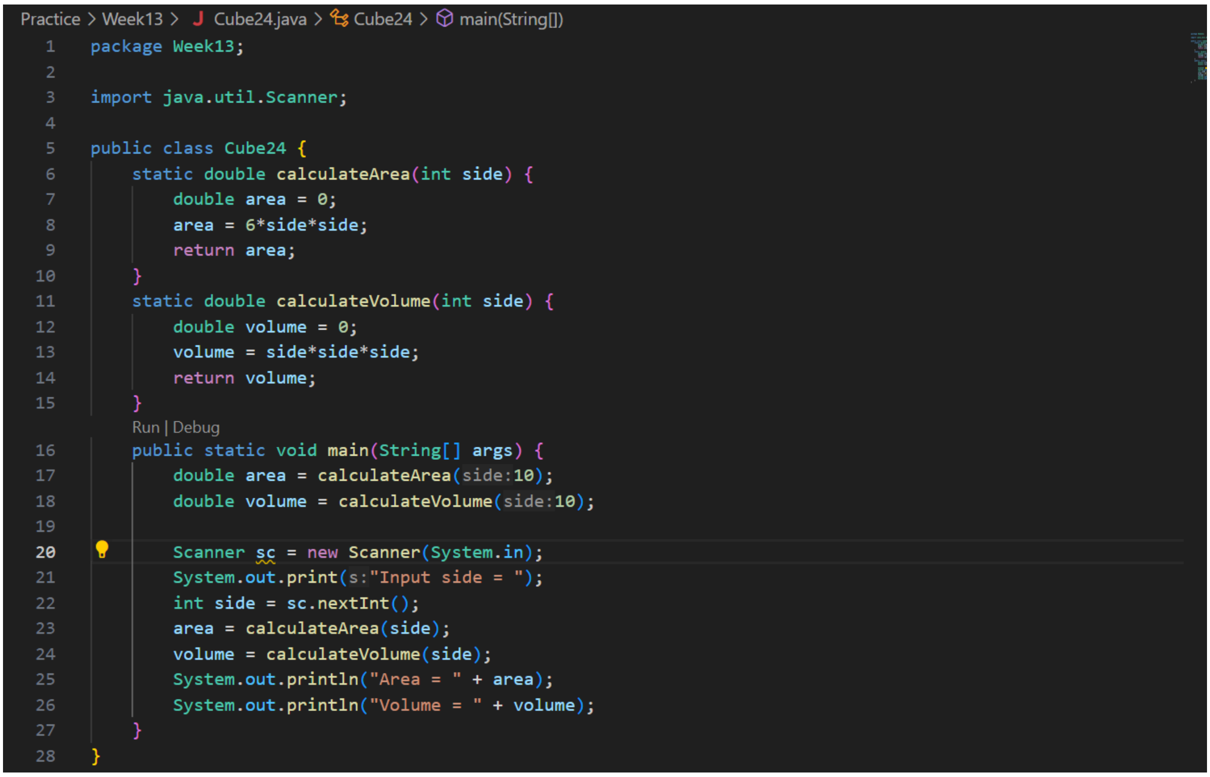
   Add functions to retrieve information from the above data with the following details:

a. Function to input students’ grade data.
b. Function to display all student grades from the first week to the seventh week.
c. Function to find the week that has the highest grade from all students.
d. Function to find the student with the highest grade (also display the grade information for each week).

3. Modify assignment number 2 above by getting the user input to determine the number of students and the number of weeks!

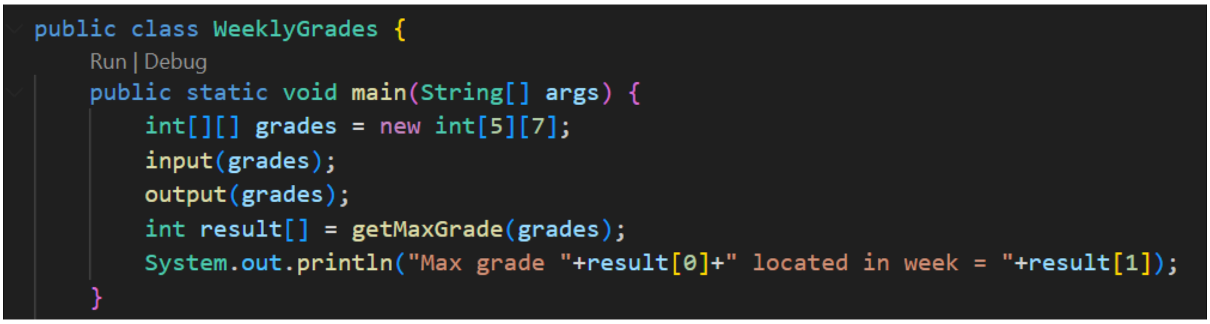
Answer!

1.



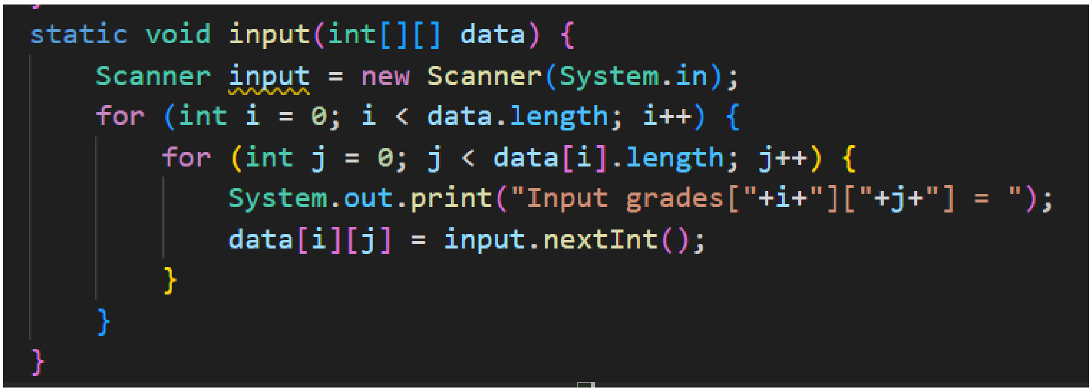
```java
package Week13;

import java.util.Scanner;

public class Cube24 {
    static double calculateArea(int side) {
        double area = 0;
        area = 6*side*side;
        return area;
    }
    static double calculateVolume(int side) {
        double volume = 0;
        volume = side*side*side;
        return volume;
    }
    public static void main(String[] args) {
        double area = calculateArea(side:10);
        double volume = calculateVolume(side:10);

        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Input side = ");
        int side = sc.nextInt();
        area = calculateArea(side);
        volume = calculateVolume(side);
        System.out.println("Area = " + area);
        System.out.println("Volume = " + volume);
    }
}
```

2.



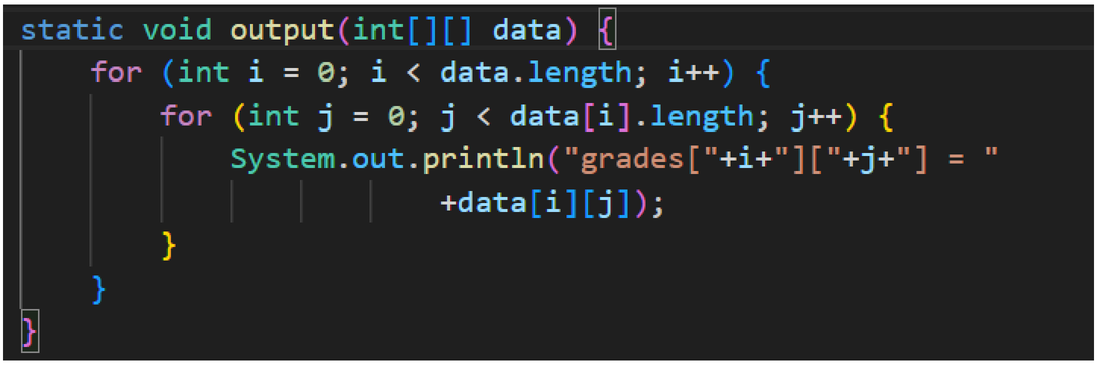
```java
public class WeeklyGrades {
    public static void main(String[] args) {
        int[][] grades = new int[5][7];
        input(grades);
        output(grades);
        int result[] = getMaxGrade(grades);
        System.out.println("Max grade "+result[0]+" located in week = "+result[1]);
    }
```

a.



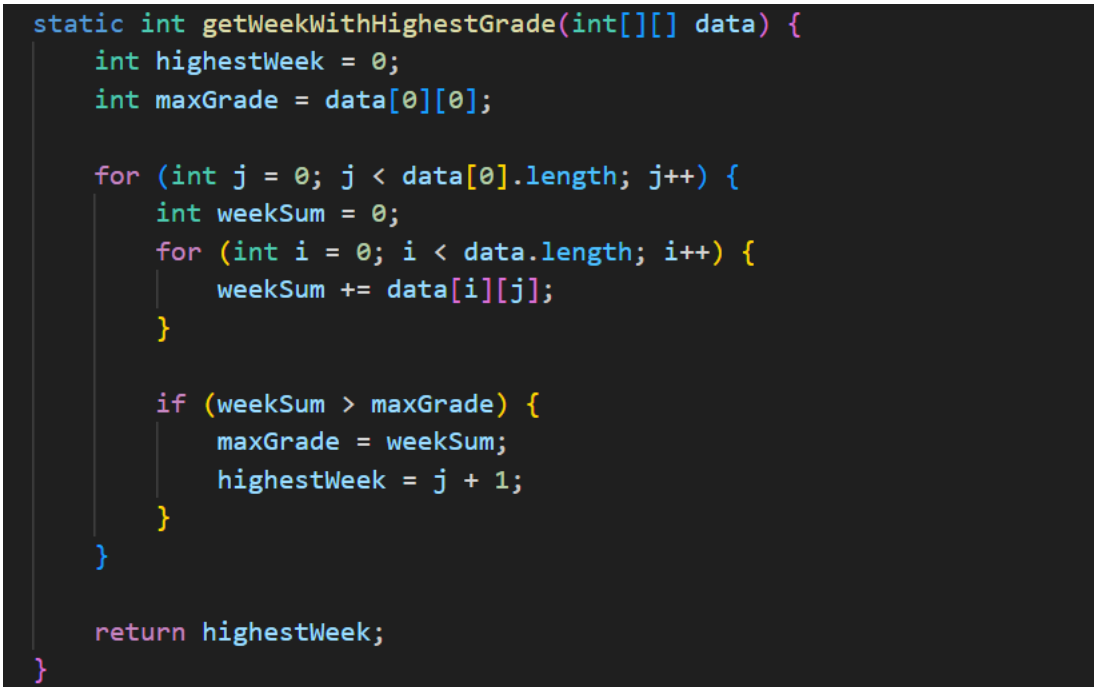
```java
static void input(int[][] data) {
    Scanner input = new Scanner(System.in);
    for (int i = 0; i < data.length; i++) {
        for (int j = 0; j < data[i].length; j++) {
            System.out.print("Input grades["+i+"]["+j+"] = ");
            data[i][j] = input.nextInt();
        }
    }
}
```

b.



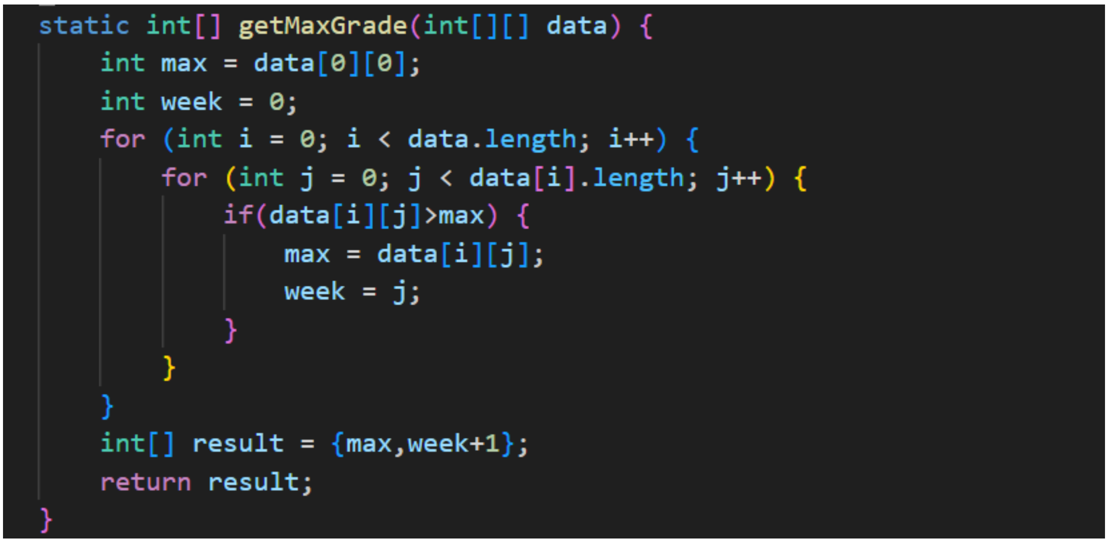
```java
static void output(int[][] data) {
    for (int i = 0; i < data.length; i++) {
        for (int j = 0; j < data[i].length; j++) {
            System.out.println("grades["+i+"]["+j+"] = "
                        +data[i][j]);
        }
    }
}
```

c.



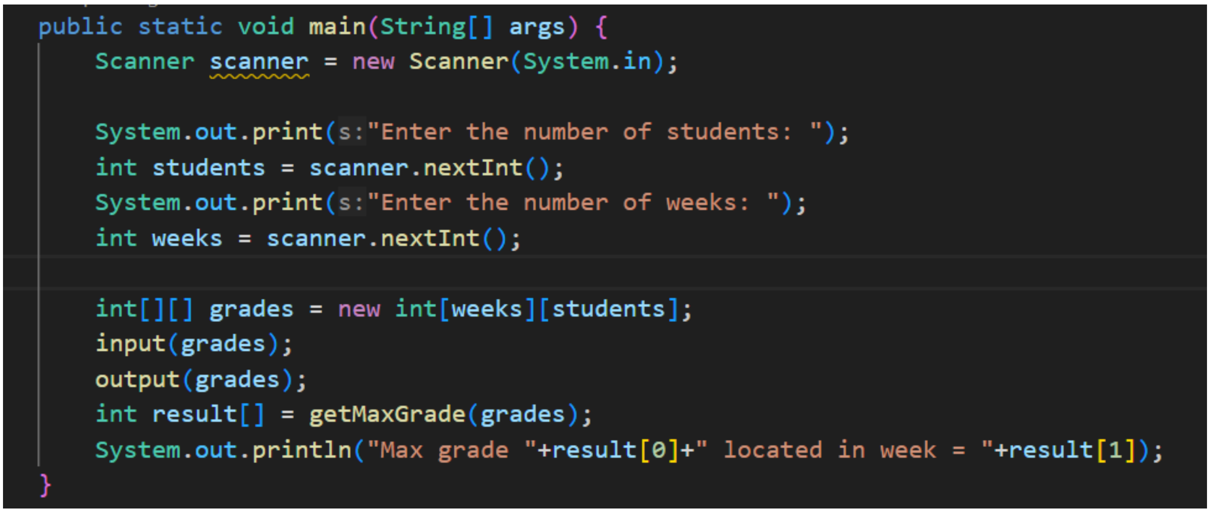
```java
static int getWeekWithHighestGrade(int[][] data) {
    int highestWeek = 0;
    int maxGrade = data[0][0];

    for (int j = 0; j < data[0].length; j++) {
        int weekSum = 0;
        for (int i = 0; i < data.length; i++) {
            weekSum += data[i][j];
        }

        if (weekSum > maxGrade) {
            maxGrade = weekSum;
            highestWeek = j + 1;
        }
    }

    return highestWeek;
}
```

d.



```
static int[] getMaxGrade(int[][] data) {
    int max = data[0][0];
    int week = 0;
    for (int i = 0; i < data.length; i++) {
        for (int j = 0; j < data[i].length; j++) {
            if(data[i][j]>max) {
                max = data[i][j];
                week = j;
            }
        }
    }
    int[] result = {max,week+1};
    return result;
}
```

3.



```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print(s:"Enter the number of students: ");
    int students = scanner.nextInt();
    System.out.print(s:"Enter the number of weeks: ");
    int weeks = scanner.nextInt();

    int[][] grades = new int[weeks][students];
    input(grades);
    output(grades);
    int result[] = getMaxGrade(grades);
    System.out.println("Max grade "+result[0]+" located in week = "+result[1]);
}
```