FARICHA AULIA

2141720155

JOBSHEET 13. Function 1

Purpose

- Students are able to understand the use of static functions in Java with parameters and returning values.
- Students are able to make programs using static functions and execute those functions.

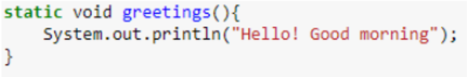
Tools and materials

- PC/Laptop
- Browsers
- Internet connection
- Anaconda3 + Java kernel (optional)

Practicum

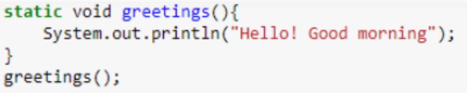
Experiment 1: Void function (does not use return value)

1. Create a function **greetings** of type void which is used to print **“Hello! Good morning"**.



```
static void greetings(){
    System.out.println("Hello! Good morning");
}
```

2. Execute or call the function **greetings**.

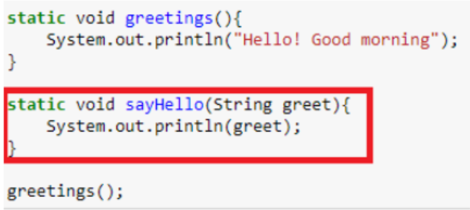


```
static void greetings(){
    System.out.println("Hello! Good morning");
}
greetings();
```

```
In [1]: // Write down the code for Experiment 1 Step 1 & 2
        static void greetings(){
            System.out.println("Hello!Good Morning");
        }
        greetings();
```

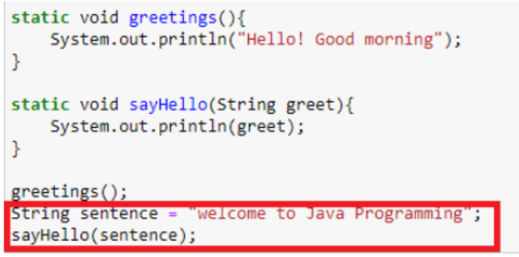
```
Hello!Good Morning
```

3. Create a **sayHello** function with a parameter of type String.



```
static void greetings(){
    System.out.println("Hello! Good morning");
}

static void sayHello(String greet){
    System.out.println(greet);
}

greetings();
```

4. Create a **greeting** variable of type String then execute or call the **sayHello** function by filling in the parameters with the **sentence** variable that has been created.



```
static void greetings(){
    System.out.println("Hello! Good morning");
}

static void sayHello(String greet){
    System.out.println(greet);
}

greetings();
String sentence = "welcome to Java Programming";
sayHello(sentence);
```

In [2]:

```
// Write down the code for Experiment 1 Step 1, 2, 3 & 4
static void greetings(){
    System.out.println("Hello! Good morning");
}
static void sayHello(String greet){
    System.out.println(greet);
}
greetings();
String sentence = "Welcome to Java Programmming";
sayHello(sentence);
```

```
Hello! Good morning
Welcome to Java Programmming
```

Question

1. Explain the differences in the functions of **greetings** and **sayHello** in practicum 1!

> Greetings uses a void function without parameters while sayHello uses a void function with parameters

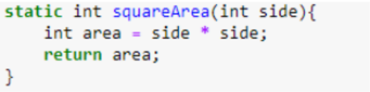
2. Explain how to call a void function with parameters and without parameters!

> Parameters are variables that function to store values to be inputted into functions. How to call using parameters:
>
> 1. Parameters are written between brackets (...);
> 2. Parameters must be assigned a data type;
> 3. If there is more than one parameter, they are separated by a comma.
>
> Creating a function without using parameters just calls the function without passing one or more. The method :
>
> 1. we create a message() function that has no argument values;
> 2. when we call the function, we don't send any value/parameter. (If there is a value that we send, it will be considered an error because in the function definition process we do not prepare an argument in the form of a variable that holds the value sent by the function call)

Experiment 2: Function with return value (Not void)

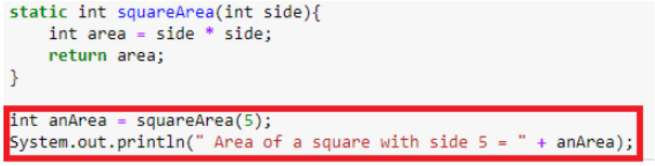
In Experiment 2, the program code created is used to calculate the area of a square by creating a function **area of a square** that uses parameters.

1. Create a **aquareArea** function to calculate the area of a square that returns the area value (int) and the side input parameter (int).



```
static int squareArea(int side){
    int area = side * side;
    return area;
}
```

2. Execute or call the squareArea function by creating a new variable, namely **area**, then fill in the variable by calling the squareArea function and filling in the side parameters. Next print the variable area to display the area of the rectangle



```
static int squareArea(int side){
    int area = side * side;
    return area;
}

int anArea = squareArea(5);
System.out.println(" Area of a square with side 5 = " + anArea);
```

In [3]:

```
// Write the code for Experiment 2 Step 1 & 2
static int squareArea(int side){
 int area = side * side;
 return area;
}
 int anArea = squareArea(5);
 System.out.println("Area of a square with side 5 = " + anArea);
```

```
Area of a square with side 5 = 25
```

Question

1. explain why when calling the **squareArea** function you have to create a new variable, namely area?

> To accommodate the result of function processing

2. Explain the use of **area return** in experiment 2 above!

> To stop processing the function and will not execute the code below

3. Modify the program in experiment 2, by making the length of **side** as input!

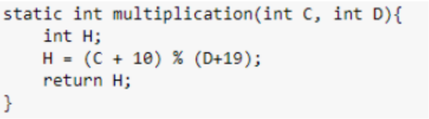
In [4]:
```
// Write down answer number 2
import java.util.Scanner;

static int squareArea(int side){
    int area = side * side;
    return area;
}

Scanner input = new Scanner(System.in);
System.out.print("Input the side = ");
int x = input.nextInt();
int anArea = squareArea(x);
System.out.println("Square Area with side " + x + " = " + anArea);
```

```
Input the side = 8
Square Area with side 8 = 64
```

Experiment 3: Function can CALL Other Functions

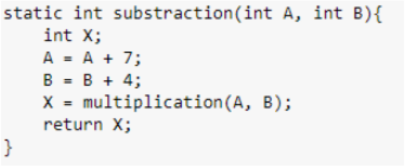
In Experiment 3, the program code created is used to implement that function can CALL other functions. Where in this experiment there is a function ** multiplication and substraction **.

1. Create a **multiplication** function that returns the value H (int) and the input parameters C and D (int).



```
static int multiplication(int C, int D){
    int H;
    H = (C + 10) % (D+19);
    return H;
}
```

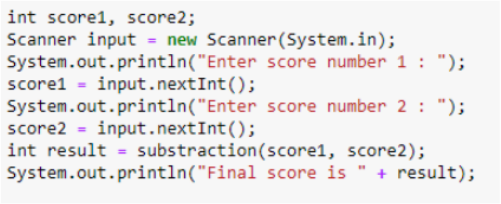
2. Create a **substraction** function that returns the value X (int) and the input parameters A and B (int) and calls the multiplication function.



```
static int substraction(int A, int B){
    int X;
    A = A + 7;
    B = B + 4;
    X = multiplication(A, B);
    return X;
}
```

3. Import the Scanner class as input in the next step.

4. Execute or call the **substraction** function.



```
int score1, score2;
Scanner input = new Scanner(System.in);
System.out.println("Enter score number 1 : ");
score1 = input.nextInt();
System.out.println("Enter score number 2 : ");
score2 = input.nextInt();
int result = substraction(score1, score2);
System.out.println("Final score is " + result);
```

In [5]:

```java
// Write the code for Experiment 3 Steps 1, 2, 3 & 4
static int multiplication(int C, int D){
    int H;
    H = (C + 10) % (D + 19);
    return H;
}

static int substraction(int A, int B){
    int X;
    A = A + 7;
    B = B + 4;
    X = multiplication(A, B);
    return X;
}

int score1, score2;
Scanner input = new Scanner(System.in);
System.out.print("Enter the Score number 1 : ");
score1 = input.nextInt();
System.out.print("Enter the Score number 2 : ");
score2 = input.nextInt();
int result = substraction(score1, score2);
System.out.println ("Final score is " + result);
```

```
Enter the Score number 1 : 13
Enter the Score number 2 : 17
Final score is 30
```

Question

1. Modify the above experiment where the **multiplication** function can call the **substraction** function then execute or call the multiplication function

In [6]:

```java
// Write down answer number 1
static int  multiplication(int C, int D){
    int H;
    C = C + 10;
    D = D + 19;
    H = substraction(C, D);
    return H;
}

static int substraction(int A, int B){
    int X;
    X = (A + 7) % (B + 4);
    return X;
}

int score1, score2;
Scanner input = new Scanner(System.in);
System.out.print("Enter the Score number 1 : ");
score1 = input.nextInt();
System.out.print("Enter the Score number 2 : ");
score2 = input.nextInt();
int result = multiplication(score1, score2);
System.out.println ("Final score is " + result);
```

```
Enter the Score number 1 : 30
Enter the Score number 2 : 8
Final score is 16
```

2. Explain the flow of the program in experiment 3 starting from input to output!

> The user is asked to input the values 1 and 2 after which the results will call the subtraction function. In the subtraction function add these 2 values then call the multiplication function. In the multiplication function there is addition and modulus then return the value and print it in the final result in the result variable

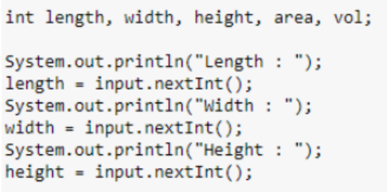
Experiment 4: Changing Programs Not Using Functions and Using Functions

In Experiment 4, the program code created is used to calculate the area of a rectangle and the volume of a block without using a function and using a function.

1. Import and declare Scanner with the name **input**

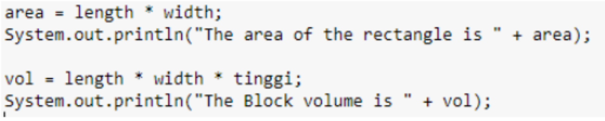
```java
import java.util.Scanner;

Scanner input = new Scanner(System.in);
```

2. Make input length, width, and height



```
int length, width, height, area, vol;

System.out.println("Length : ");
length = input.nextInt();
System.out.println("Width : ");
width = input.nextInt();
System.out.println("Height : ");
height = input.nextInt();
```

3. Calculate the area of the rectangle and the volume of the block



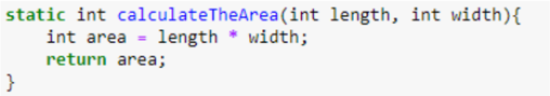
```
area = length * width;
System.out.println("The area of the rectangle is " + area);

vol = length * width * tinggi;
System.out.println("The Block volume is " + vol);
```

```
In [7]: // Write the code for Experiment 4 Steps 1, 2, & 3
import java.util.Scanner;
Scanner input = new Scanner(System.in);

int length,width,height,area,vol;

System.out.print("Length : ");
length = input.nextInt();
System.out.print("Width : ");
width = input.nextInt();
System.out.print("Height : ");
height = input.nextInt();

area = length * width;
System.out.println("The area of rectangle is : " + area);
vol = length * width * height;
System.out.println("The block volume is : " + vol);
```

```
Length : 8
Width : 4
Height : 6
The area of rectangle is : 32
The block volume is : 192
```

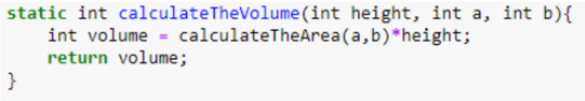
4. The program calculates the area of the square and the volume of the block above, if a function is made, there are 3 functions, namely calculateArea, calculateVolume and the main function, as below:

Function calculateArea



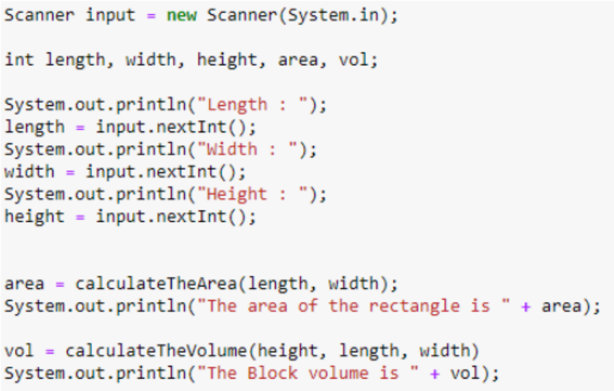
```
static int calculateTheArea(int length, int width){
    int area = length * width;
    return area;
}
```

CalculateVolume . function



```
static int calculateTheVolume(int height, int a, int b){
    int volume = calculateTheArea(a,b)*height;
    return volume;
}
```

5. Eksekusi/panggil fungsi **calculateArea** dan **calculateVolume**



```
Scanner input = new Scanner(System.in);

int length, width, height, area, vol;

System.out.println("Length : ");
length = input.nextInt();
System.out.println("Width : ");
width = input.nextInt();
System.out.println("Height : ");
height = input.nextInt();


area = calculateTheArea(length, width);
System.out.println("The area of the rectangle is " + area);

vol = calculateTheVolume(height, length, width)
System.out.println("The Block volume is " + vol);
```

In [8]:

```java
// Write the code for Experiment 4 Steps 1, 2, & 3
static int calculateTheArea(int length, int width){
 int area = length*width;
 return area;
}
static int calculateTheVolume(int height, int a,int b){
 int volume = calculateTheArea(a,b)*height;
 return volume;
}
import java.util.Scanner;
Scanner input = new Scanner(System.in);
int length,width,height,area,vol;

System.out.print("Length : ");
length = input.nextInt();
System.out.print("Width : ");
width = input.nextInt();
System.out.print("Height : ");
height = input.nextInt();

area = calculateTheArea(length, width);
System.out.println("The area of rectangle is : " + area);
vol = calculateTheVolume(height, length, width);
System.out.println("The block volume is : " + vol);
```

```
Length : 8
Width : 6
Height : 3
The area of rectangle is : 48
The block volume is : 144
```

Question

1. Explain the use of the parameters contained in the functions calculateArea and calculeVolume!

> The use of parameters in both functions is to add up the values of the input to be added up in the function then return it and call it below with a new variable

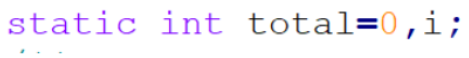
2. After doing experiment 4, in your opinion, which program is more efficient, using a function or without a function? Explain!

> Without a function, because if you use a function it will add a lot of code and it is not efficient to use it quickly and it will be more efficient if it is used to make a long code and if you want to search it is only changed in the function section

Experiment 5: Functions Using Arrays and Global Variables

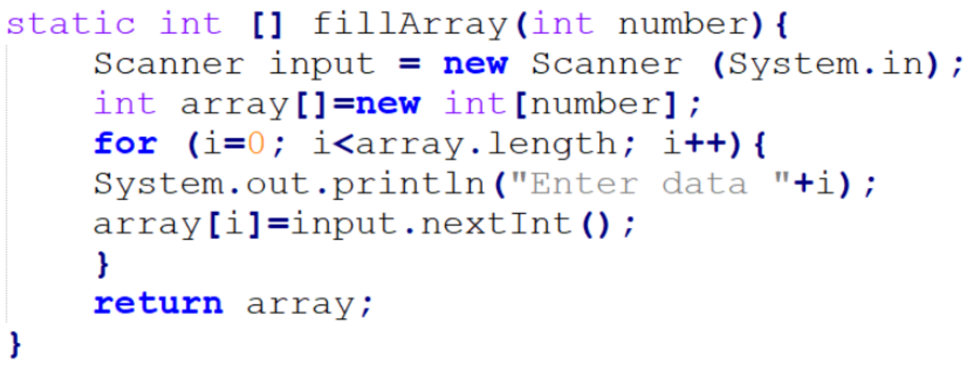
In Experiment 5, the program code created is used to calculate the total values in the array by creating 3 functions, namely fillArray, calcTol, and showArray.

1. Make **global variable total and i** of type int



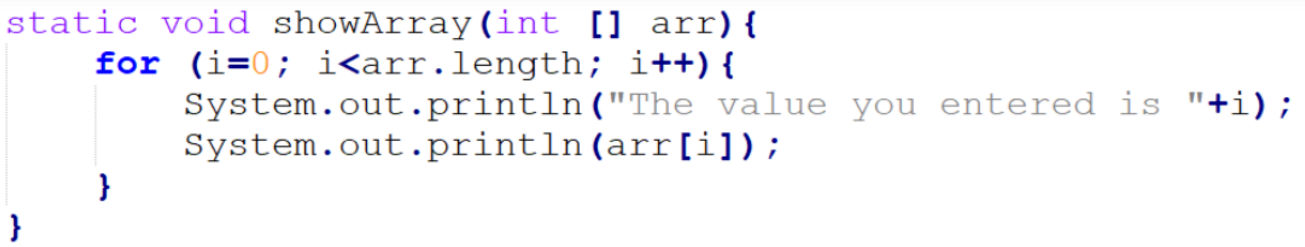
```java
static int total=0,i;
```

2. Create a function **fillArray** of type int with a number parameter of type int



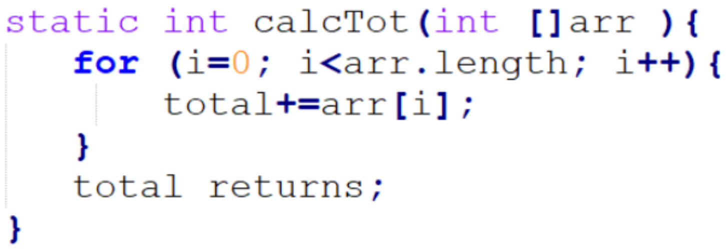
```java
static int [] fillArray(int number){
    Scanner input = new Scanner (System.in);
    int array[]=new int[number];
    for (i=0; i<array.length; i++){
    System.out.println("Enter data "+i);
    array[i]=input.nextInt();
    }
    return array;
}
```

3. Create a function **showArray** of type **void** with array data parameter **arr** of type int



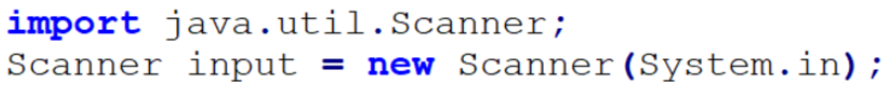
```
static void showArray(int [] arr){
    for (i=0; i<arr.length; i++){
        System.out.println("The value you entered is "+i);
        System.out.println(arr[i]);
    }
}
```

4. Create a function **calcTot** of type int with array data parameter **arr** of type int



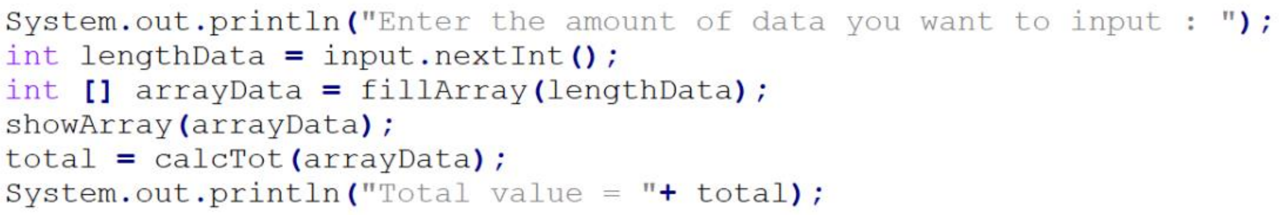
```
static int calcTot(int []arr ){
    for (i=0; i<arr.length; i++){
        total+=arr[i];
    }
    total returns;
}
```

5. Import and declare Scanner with the name **input**



```
import java.util.Scanner;
Scanner input = new Scanner(System.in);
```

6. Execute or call the three functions, namely **fillArray, displayArray, and calcTot**, then run the program!



```
System.out.println("Enter the amount of data you want to input : ");
int lengthData = input.nextInt();
int [] arrayData = fillArray(lengthData);
showArray(arrayData);
total = calcTot(arrayData);
System.out.println("Total value = "+ total);
```

In [9]:

```
// Write the code for Experiment 4 Steps 1 to 6
static int total = 0,i;

static int [] fillArray (int number){
    Scanner input = new Scanner(System.in);
    int array[] = new int[number];
    for (i = 0; i < array.length; i++){
    System.out.print("Enter data " + (i+1) + " : ");
    array[i] = input.nextInt();
    }
    return array;
}

static void showArray (int [] arr){
    for(i = 0; i < arr.length; i++){
        System.out.print("The value you entered is " + (i+1) + " : ");
        System.out.println(arr[i]);
    }
}

static int calcTot (int [] arr){
    for(i = 0; i < arr.length; i++){
    total+=arr[i];
    }
    return total;
}

import java.util.Scanner;
Scanner input = new Scanner(System.in);
System.out.print("Enter the amount the data you want to input : ");
int lengthData = input.nextInt();
int []arrayData = fillArray(lengthData);
showArray(arrayData);
total = calcTot(arrayData);
System.out.println("Total value : " + total);
```

```
Enter the amount the data you want to input : 3
Enter data 1 : 2
Enter data 2 : 4
Enter data 3 : 6
The value you entered is 1 : 2
The value you entered is 2 : 4
The value you entered is 3 : 6
Total value : 12
```

Question

1. Explain why the return function array is made of type void, while fillArray and calcTot are of type int!

> Because the array display does not require a return value while fillArray is a sum so it requires a return value so that it can be output

2. In your opinion, can the fillArray and calcTot function be replaced with void type? Explain and prove it with the program!

> Can't, because calcTot and fillArray require a return value and they are of type so they will error when executed

In [10]:

```
// Write down answer number 2
static int total = 0,i;

static void [] fillArray (int number){
    Scanner input = new Scanner(System.in);
    int array[] = new int[number];
    for (i = 0; i < array.length; i++){
    System.out.print("Enter data " + (i+1) + " : ");
    array[i] = input.nextInt();
    }
}

static void showArray (int [] arr){
    for(i = 0; i < arr.length; i++){
        System.out.print("The value you entered is " + (i+1) + " : ");
        System.out.println(arr[i]);
    }
}

static void calcTot (int [] arr){
    for(i = 0; i < arr.length; i++){
    total+=arr[i];
    }
}

import java.util.Scanner;
Scanner input = new Scanner(System.in);
System.out.print("Enter the amount the data you want to input : ");
int lengthData = input.nextInt();
int []arrayData = fillArray(lengthData);
showArray(arrayData);
total = calcTot(arrayData);
System.out.println("Total value : " + total);
```

```
|   static void [] fillArray (int number){
illegal start of expression

|   static void [] fillArray (int number){
';' expected

|   static void [] fillArray (int number){
'.class' expected

|   static void [] fillArray (int number){
';' expected

|   static void [] fillArray (int number){
unexpected type
  required: value
  found:    class

|       int array[] = new int[number];
cannot find symbol
  symbol:   variable number
```

Task

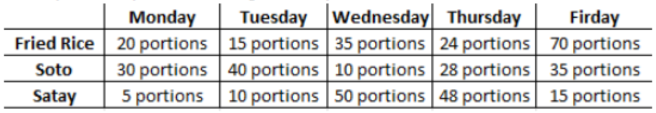
1. Create a static method named Max3(int number1, int number2, int number3) which accepts 3 integer parameters and returns an integer which is the maximum value among the three numbers.

In [11]:

```java
// Write down answer number 1
static int Max3(int num1, int num2, int num3){
    int array[] = {num1, num2, num3};
    int max = 0;

for (int i = 0; i < array.length; i++){
    if (array[i] > max){
    max = array[i];
    }
}
return max;
}

import java.util.Scanner;
Scanner input = new Scanner(System.in);
int v1, v2, v3;
System.out.print("Enter value 1 : ");
v1 = input.nextInt();
System.out.print("Enter value 2 : ");
v2 = input.nextInt();
System.out.print("Enter value 3 : ");
v3 = input.nextInt();
int vMax = Max3(v1, v2, v3);
System.out.println("Maximum Value of the three values : " + vMax);
```

```
Enter value 1 : 9
Enter value 2 : 6
Enter value 3 : 8
Maximum Value of the three values : 9
```

2. In a restaurant there are 3 menus that are sold, namely fried rice, soto, and satay. The price of fried rice is Rp. 20,000, soto Rp. 15,000, and satay Rp. 25,000. The restaurant is open from Monday to Friday. The following is a daily sales table for each menu at the restaurant from Monday to Friday



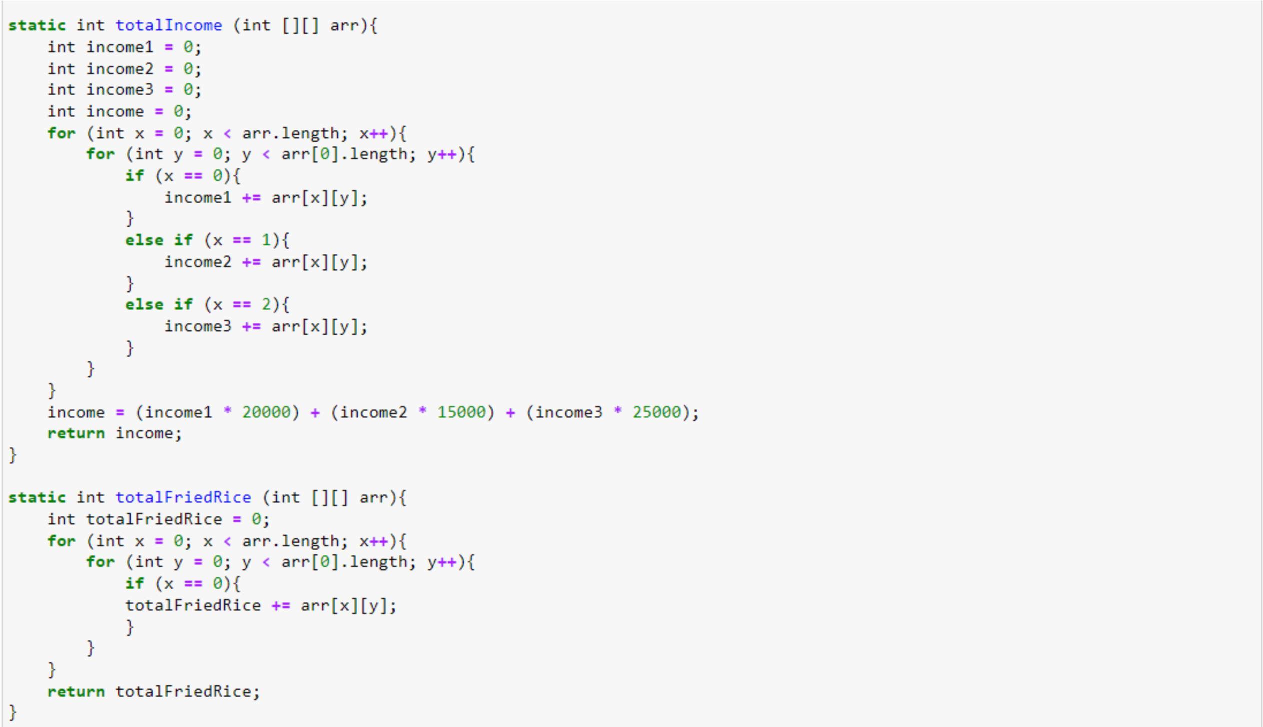
	Monday	Tuesday	Wednesday	Thursday	Firday
Fried Rice	20 portions	15 portions	35 portions	24 portions	70 portions
Soto	30 portions	40 portions	10 portions	28 portions	35 portions
Satay	5 portions	10 portions	50 portions	48 portions	15 portions

Create a Function as follows:

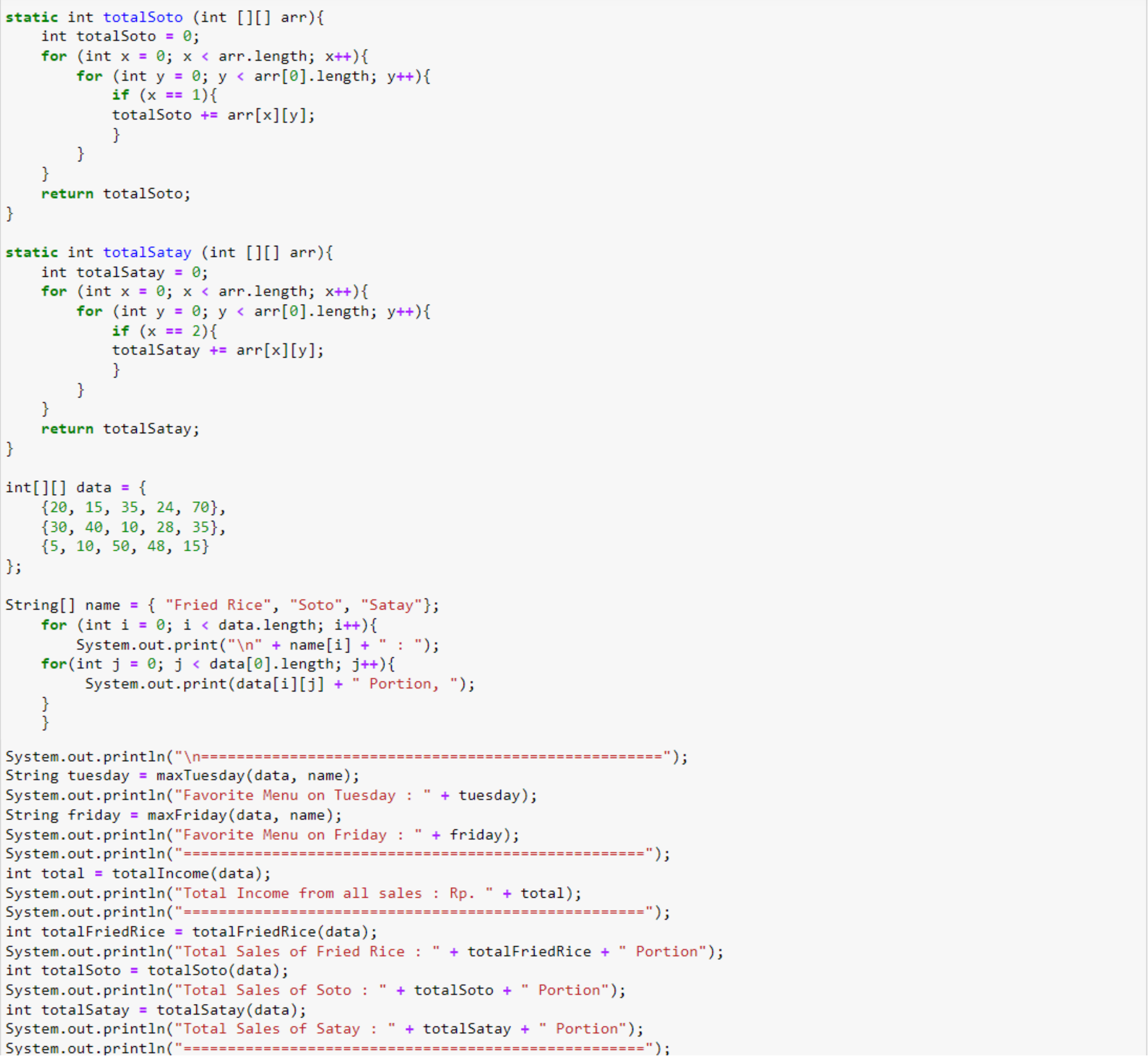
- Function to display favorite menu on Tuesday and Friday (favorite menu is assumed to be the most sold menu on that day)
- Function to calculate the restaurant's income from Monday to Friday.
- Function to calculate how many servings are sold for each menu, namely fried rice, soto, and satay from Monday to Friday.

In [12]:

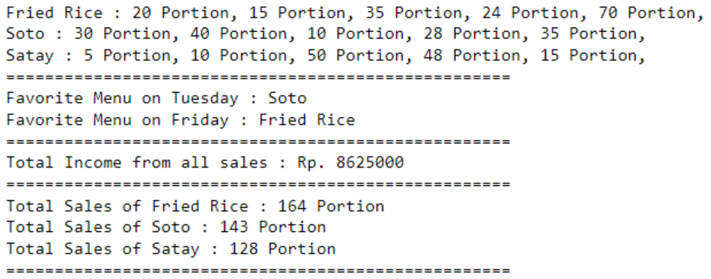
```java
// Write down answer number 2
import java.util.Scanner;

static String maxTuesday (int [][] arr, String [] arg){
    String food = "";
    int max = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (arr[x][1] > max){
                max = arr[x][y];
                food = arg[x];
            }
        }
    }
    return food;
}

static String maxFriday (int [][] arr, String [] arg){
    String food = "";
    int max = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (arr[x][4] > max){
                max = arr[x][y];
                food = arg[x];
            }
        }
    }
    return food;
}
```



```java
static int totalIncome (int [][] arr){
    int income1 = 0;
    int income2 = 0;
    int income3 = 0;
    int income = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (x == 0){
                income1 += arr[x][y];
            }
            else if (x == 1){
                income2 += arr[x][y];
            }
            else if (x == 2){
                income3 += arr[x][y];
            }
        }
    }
    income = (income1 * 20000) + (income2 * 15000) + (income3 * 25000);
    return income;
}

static int totalFriedRice (int [][] arr){
    int totalFriedRice = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (x == 0){
            totalFriedRice += arr[x][y];
            }
        }
    }
    return totalFriedRice;
}
```



```java
static int totalSoto (int [][] arr){
    int totalSoto = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (x == 1){
            totalSoto += arr[x][y];
            }
        }
    }
    return totalSoto;
}

static int totalSatay (int [][] arr){
    int totalSatay = 0;
    for (int x = 0; x < arr.length; x++){
        for (int y = 0; y < arr[0].length; y++){
            if (x == 2){
            totalSatay += arr[x][y];
            }
        }
    }
    return totalSatay;
}

int[][] data = {
    {20, 15, 35, 24, 70},
    {30, 40, 10, 28, 35},
    {5, 10, 50, 48, 15}
};

String[] name = { "Fried Rice", "Soto", "Satay"};
    for (int i = 0; i < data.length; i++){
        System.out.print("\n" + name[i] + " : ");
    for(int j = 0; j < data[0].length; j++){
         System.out.print(data[i][j] + " Portion, ");
    }
    }

System.out.println("\n====================================================");
String tuesday = maxTuesday(data, name);
System.out.println("Favorite Menu on Tuesday : " + tuesday);
String friday = maxFriday(data, name);
System.out.println("Favorite Menu on Friday : " + friday);
System.out.println("====================================================");
int total = totalIncome(data);
System.out.println("Total Income from all sales : Rp. " + total);
System.out.println("====================================================");
int totalFriedRice = totalFriedRice(data);
System.out.println("Total Sales of Fried Rice : " + totalFriedRice + " Portion");
int totalSoto = totalSoto(data);
System.out.println("Total Sales of Soto : " + totalSoto + " Portion");
int totalSatay = totalSatay(data);
System.out.println("Total Sales of Satay : " + totalSatay + " Portion");
System.out.println("====================================================");
```



```
Fried Rice : 20 Portion, 15 Portion, 35 Portion, 24 Portion, 70 Portion,
Soto : 30 Portion, 40 Portion, 10 Portion, 28 Portion, 35 Portion,
Satay : 5 Portion, 10 Portion, 50 Portion, 48 Portion, 15 Portion,
====================================================
Favorite Menu on Tuesday : Soto
Favorite Menu on Friday : Fried Rice
====================================================
Total Income from all sales : Rp. 8625000
====================================================
Total Sales of Fried Rice : 164 Portion
Total Sales of Soto : 143 Portion
Total Sales of Satay : 128 Portion
====================================================
```