

Basic Programming Practicum

Function 1



Name

Dicha Zelianivan Arkana

NIM

2241720002

Class

1i

Department

Information Technology

Study Program

D4 Informatics Engineering

Contents

1	Laboratory	2
1.1	Experiment 1	2
1.2	Experiment 2	3
1.3	Experiment 3	4
1.4	Experiment 4	5
1.5	Experiment 5	7
1.6	Experiment 6	8
2	Questions!	10
3	Assignment	12

1 Laboratory

1.1 Experiment 1

1. Create a new project
2. Create a new class, name it `Greeting`
3. Create a function called `giveGreeting` inside the class

```
public class Greeting {  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
}
```

4. Create a `main` function inside the class, and execute the `giveGreeting` function from within the `main` function.

```
public class Greeting {  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
    }  
}
```

5. Compile and run the program

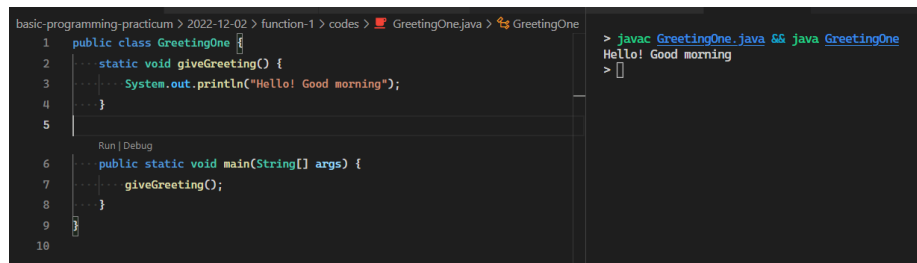
A screenshot of an IDE window. The left pane shows the source code for a Java class named GreetingOne. The code includes a static method giveGreeting() that prints "Hello! Good morning" and a main method that calls giveGreeting(). The right pane shows the command prompt with the command 'javac GreetingOne.java' and 'java GreetingOne' executed, resulting in the output 'Hello! Good morning'.

Figure 1: Experiment 1 code and output

1.2 Experiment 2

1. Using the class that was created in Experiment 1, add function called saySomething inside the Greeting class

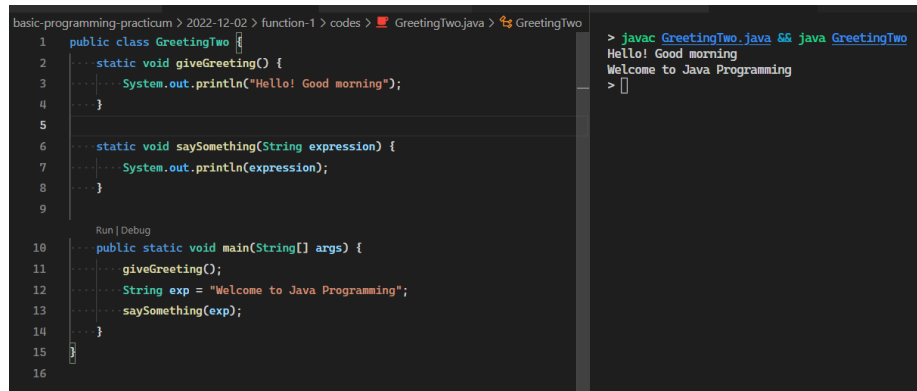
```
public class Greeting {  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    static void saySomething(String expression) {  
        System.out.println(expression);  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
    }  
}
```

2. Execute the saySomething function from inside the main function

```
public class Greeting {  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    static void saySomething(String expression) {  
        System.out.println(expression);  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
    }  
}
```

```
        String exp = "Welcome to Java Programming";
        saySomething(exp);
    }
}
```

3. Compile and run the program



The screenshot shows an IDE with a dark theme. On the left, the code for `GreetingTwo.java` is displayed. It defines a class `GreetingTwo` with two static methods: `giveGreeting()` which prints "Hello! Good morning", and `saySomething(String expression)` which prints the given expression. A `main` method is also present, which calls `giveGreeting()`, sets a string `exp` to "Welcome to Java Programming", and calls `saySomething(exp)`. On the right, the terminal output shows the command `> javac GreetingTwo.java && java GreetingTwo` being executed, followed by the printed output: "Hello! Good morning" and "Welcome to Java Programming".

Figure 2: Experiment 2 code and output

1.3 Experiment 3

1. Create a new class, name it `Square`
2. Create a function named `squareArea` inside that class which returns the value area (int), with the input parameter side (int)

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

3. Create a `main` function inside the class, and execute the `squareArea` function from within the `main` function.

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

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

```

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

```

4. Compile and run the program

The screenshot shows an IDE with a dark theme. The left pane displays the source code for `Square.java`. The code defines a `squareArea` method and a `main` method. The right pane shows the command prompt output after running the program.

```

basic-programming-practicum > 2022-12-02 > function-1 > codes > Square.java > Square
1 public class Square {
2     static int squareArea(int side) {
3         int area = side * side;
4         return area;
5     }
6
7     public static void main(String[] args) {
8         int a = squareArea(5);
9         System.out.println("Area of a square with side = 5 is " + a);
10    }
11
12
> javac Square.java && java Square
Area of a square with side = 5 is 25
>

```

Figure 3: Experiment 3 code and output

1.4 Experiment 4

1. Create a new class, name it `ArithmeticOperation`
2. Create a function named `multiplication` inside that class which returns the value `H` (int) and input parameters `C` and `D` (int)

```

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

```

3. Create a function called `substraction` inside that class which returns the value `X` (int) and input parameters `A` and `B` (int) and calls the `multiplication` function.

```

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

```

```

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

```

4. Create a main function inside the class, and execute the subtraction function from within the main function. Don't forget to add the **Scanner** library.

```

public static void main(String[] args) {
    int value1, value2;
    Scanner input = new Scanner(System.in);
    System.out.print("Input value 1: ");
    value1 = input.nextInt();
    System.out.print("Input value 2: ");
    value2 = input.nextInt();
    int result = subtraction(value1, value2);
    System.out.println("The result is " + result);
}

```

5. Compile and run the program.

The screenshot shows an IDE with two panels. The left panel displays the source code for `ArithmeticOperation.java`, which includes a `multiplication` method, a `subtraction` method, and a `main` method that uses a `Scanner` to take user input and calls the `subtraction` method. The right panel shows the terminal output after running the program, displaying the prompts and the final result of 27.

```

2022-12-02 > function-1 > codes > ArithmeticOperation.java > ArithmeticOperation > main(String[])
1  import java.util.Scanner;
2
3  public class ArithmeticOperation {
4      static int multiplication(int C, int D) {
5          int M;
6          M = (C + 10) * (D + 10);
7          return M;
8      }
9
10     static int subtraction(int A, int B) {
11         int X;
12         A = A + 7;
13         B = B + 4;
14         X = multiplication(A, B);
15         return X;
16     }
17
18     public static void main(String[] args) {
19         int value1, value2;
20         Scanner input = new Scanner(System.in);
21         System.out.print("Input value 1: ");
22         value1 = input.nextInt();
23         System.out.print("Input value 2: ");
24         value2 = input.nextInt();
25         int result = subtraction(value1, value2);
26         System.out.println("The result is " + result);
27     }
28 }
29
> javac ArithmeticOperation.java && java ArithmeticOperation
Input value 1: 10
Input value 2: 10
The result is 27
>

```

Figure 4: Experiment 4 code and output

1.5 Experiment 5

1. Create a new class, name it MultiParameter
2. Create a function called Print (void) inside the class using two types of parameter data, namely String and int

```
public class MultiParameter {  
    static void Print(String str, int... a) {  
        System.out.println("String: " + str);  
        System.out.println("Number of parameters: " + a.length);  
        for (int i : a) {  
            System.out.print(i + " ");  
        }  
        System.out.println("");  
    }  
}
```

3. Create a main function inside the class, and execute the Print function from within the main function.

```
public static void main(String[] args) {  
    Print("Basic Programming", 85, 90);  
    Print("Information Technology", 1, 2, 3, 4, 5);  
    Print("Politeknik Negeri Malang");  
}
```

4. Compile and run the program

```
programming-practicum > 2022-12-02 > function-1 > codes > MultiParameter.java > MultiParameter  
1 public class MultiParameter {  
2     static void Print(String str, int... a) {  
3         System.out.println("String: " + str);  
4         System.out.println("Number of parameters: " + a.length);  
5         for (int i : a) {  
6             System.out.println(i + " ");  
7         }  
8         System.out.println("");  
9     }  
10 }  
  
Run | Debug  
11 public static void main(String[] args) {  
12     Print(str: "Basic Programming", ...a: 85, 90);  
13     Print(str: "Information Technology", ...a: 1, 2, 3, 4, 5);  
14     Print(str: "Politeknik Negeri Malang");  
15 }  
16 }  
17 }  
  
> javac MultiParameter.java && java MultiParameter  
String: Basic Programming  
Number of parameters: 2  
85  
90  
  
String: Information Technology  
Number of parameters: 5  
1  
2  
3  
4  
5  
  
String: Politeknik Negeri Malang  
Number of parameters: 0  
> []
```

Figure 5: Experiment 5 code and output

1.6 Experiment 6

1. Create a new class, name it Geometry1
2. Create a program to calculate the area of a rectangle and volume of blocks without using functions

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int length, width, height, area, volume;
    System.out.print("Enter a length value: ");
    length = input.nextInt();
    System.out.print("Enter a width value: ");
    width = input.nextInt();
    System.out.print("Enter a height value: ");
    height = input.nextInt();
    area = length * width;
    System.out.println("Area of rectangle is " + area);
    volume = length * width * height;
    System.out.println("Volume of block is " + volume);
}
```

3. Create another new class, name it Geometry2
4. Geometry2 contains the program code for calculating the area of a rectangle and the volume of a block by using a function, so that there are three functions, namely calculateArea, calculateVolume, and the main function.

- calculateArea function

```
static int calculateArea(int lgt, int wdt) {
    int a = lgt * wdt;
    return a;
}
```

- calculateVolume function

```
static int calculateVolume(int hgt, int a, int b) {
    int vol = calculateArea(a, b) * hgt;
    return vol;
}
```

- main function

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int length, width, height, area, volume;
    System.out.print("Enter a length value: ");
    length = input.nextInt();
```

```

System.out.print("Enter a width value: ");
width = input.nextInt();
System.out.print("Enter a height value: ");
height = input.nextInt();
area = calculateArea(length, width);
System.out.println("Area of rectangle is " + area);
volume = calculateVolume(height, length, width);
System.out.println("Volume of block is " + volume);
}

```

5. Compile and run the two programs (class Geometry1 and Geometry2)

```

basic-programming-practicum > 2022-12-02 > function-1 > codes > Geometry1.java > ...
1  import java.util.Scanner;
2
3  public class Geometry1 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6          int length, width, height, area, volume;
7          System.out.print("Enter a length value: ");
8          length = input.nextInt();
9          System.out.print("Enter a width value: ");
10         width = input.nextInt();
11         System.out.print("Enter a height value: ");
12         height = input.nextInt();
13         area = length * width;
14         System.out.println("Area of rectangle is " + area);
15         volume = length * width * height;
16         System.out.println("Volume of block is " + volume);
17     }
18 }
19
> javac Geometry1.java && java Geometry1
Enter a length value: 2
Enter a width value: 2
Enter a height value: 2
Area of rectangle is 4
Volume of block is 8
>

```

Figure 6: Experiment 6 Geometry1 code and output

```

basic-programming-practicum > 2022-12-02 > function-1 > codes > Geometry2.java > Geometry2
1  import java.util.Scanner;
2
3  public class Geometry2 {
4      static int calculateArea(int lgt, int wdt) {
5          int a = lgt * wdt;
6          return a;
7      }
8
9      static int calculateVolume(int hgt, int a, int b) {
10         int vol = calculateArea(a, b) * hgt;
11         return vol;
12     }
13
14     public static void main(String[] args) {
15         Scanner input = new Scanner(System.in);
16         int length, width, height, area, volume;
17         System.out.print("Enter a length value: ");
18         length = input.nextInt();
19         System.out.print("Enter a width value: ");
20         width = input.nextInt();
21         System.out.print("Enter a height value: ");
22         height = input.nextInt();
23         area = calculateArea(length, width);
24         System.out.println("Area of rectangle is: " + area);
25         volume = calculateVolume(height, length, width);
26         System.out.println("Volume of block is " + volume);
27     }
28 }
29
> javac Geometry2.java && java Geometry2
Enter a length value: 2
Enter a width value: 2
Enter a height value: 2
Area of rectangle is: 4
Volume of block is 8
>

```

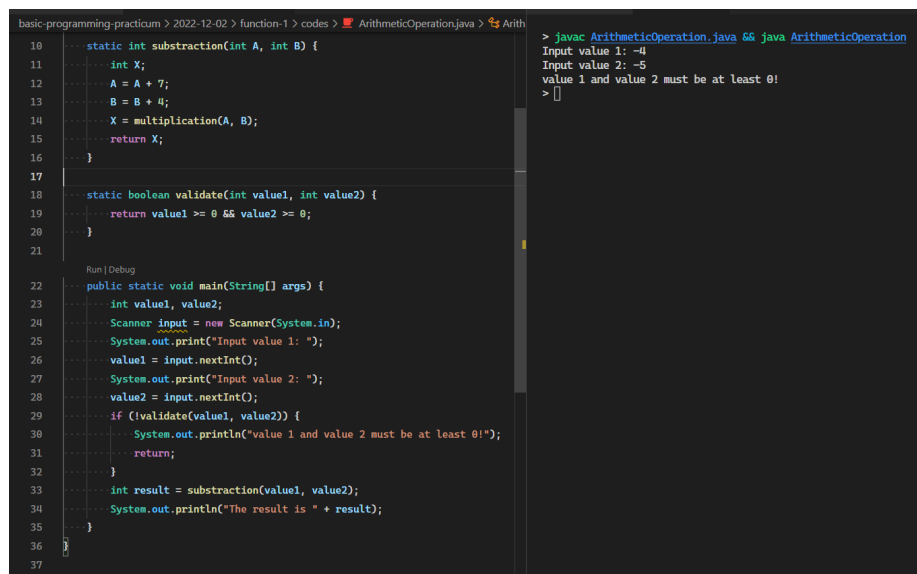
Figure 7: Experiment 6 Geometry2 code and output

6. Describe the flow of the program for calculating the area of a rectangle and volume of blocks in class Geometry2

- The length, width, and height is asked from the user using Scanner
- After inputting those values, it calculates the area using the method `calculateArea` that has been declared before
- Inside the `calculateArea`, the width and height is multiplied
- When the area has been calculated, it outputs the value
- Next, the volume is calculated using the `calculateVolume` method that has been declared before
- Inside the `calculateVolume` method, it finds the area using `calculateArea` method and multiply it with the height
- It outputs the volume after it being calculated

2 Questions!

1. Based on experiment 2 and 3, explain when a function requires a return value!
A function should return a value when the caller of the function wants a value from calling the function.
2. In Experiment 4, add a function that is used to ensure that the `value1` and `value2` are at least 0, then call that function in the `main`!



```
basic-programming-practicum > 2022-12-02 > function-1 > codes > ArithmeticOperation.java > Arith
10 //static int subtraction(int A, int B) {
11 //    int X;
12 //    A = A + 7;
13 //    B = B + 4;
14 //    X = multiplication(A, B);
15 //    return X;
16 //}
17
18 //static boolean validate(int value1, int value2) {
19 //    return value1 >= 0 && value2 >= 0;
20 //}
21
22 //Run | Debug
23 //public static void main(String[] args) {
24 //    int value1, value2;
25 //    Scanner input = new Scanner(System.in);
26 //    System.out.print("Input value 1: ");
27 //    value1 = input.nextInt();
28 //    System.out.print("Input value 2: ");
29 //    value2 = input.nextInt();
30 //    if (!validate(value1, value2)) {
31 //        System.out.println("value 1 and value 2 must be at least 0!");
32 //        return;
33 //    }
34 //    int result = subtraction(value1, value2);
35 //    System.out.println("The result is " + result);
36 //}
37
> javac ArithmeticOperation.java && java ArithmeticOperation
Input value 1: -4
Input value 2: -5
value 1 and value 2 must be at least 0!
> []
```

Figure 8: Question 2 code and output

-
3. Explain why the parameter entries in Experiment 5 are written with `int...` a!

It's a syntax for *variable arguments* or *varargs* for short. It is used so that we can pass in an arbitrary number of arguments into the function and the function will collect it as an array that can be used inside the function.

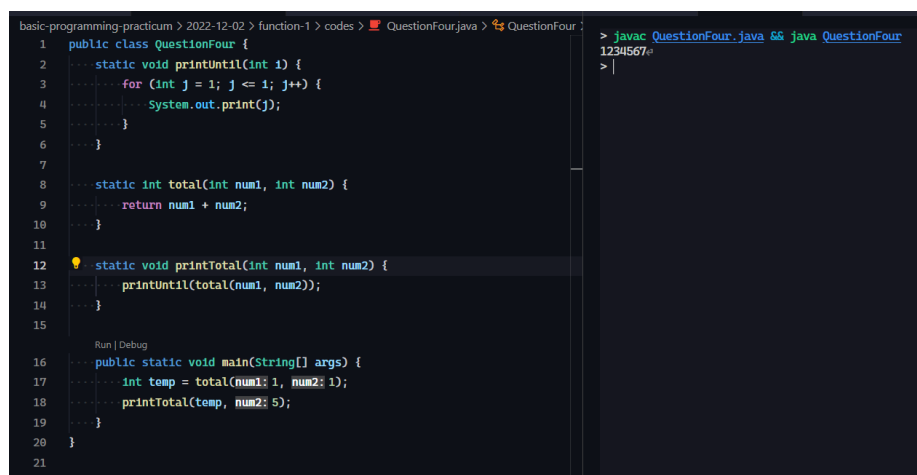
4. What is the output of the program below, then explain the flow of the program!

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

    static int total(int num1, int num2) {
        return num1 + num2;
    }

    static void printTotal(int num1, int num2) {
        printUntil(total(num1, num2));
    }

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



The screenshot shows an IDE with a code editor on the left and a terminal on the right. The code editor contains the following Java code:

```
1 public class QuestionFour {
2     static void printUntil(int i) {
3         for (int j = 1; j <= i; j++) {
4             System.out.print(j);
5         }
6     }
7
8     static int total(int num1, int num2) {
9         return num1 + num2;
10    }
11
12    static void printTotal(int num1, int num2) {
13        printUntil(total(num1, num2));
14    }
15
16    public static void main(String[] args) {
17        int temp = total(num1: 1, num2: 1);
18        printTotal(temp, num2: 5);
19    }
20 }
21
```

The terminal on the right shows the command `> javac QuestionFour.java && java QuestionFour` and the output `1234567`.

Figure 9: Question 4 code and output

-
- Starting from main, it will call the `total` function with an argument of 1 and 1
 - Inside the `total` function, it will sum `num1` and `num2`
 - In this case, the return value is an integer of 2 because $1 + 1 = 2$
 - After getting the value from `total`, it calls the `printTotal` function with an argument of `temp` and 5
 - Inside the `printTotal` function, it will invoke the `printUntil` function with an argument of `total(num1, num2)`
 - `total(num1, num2)` will result in 7 because `num1` will be 2 and `num2` will be 5
 - The `printUntil` function will print the value of `j` for `i`-many times, in this case 7 times

3 Assignment

1. Create a static method called `Max3(int bil1, int bil2, int bil3)` which takes three integer parameters and returns an integer number which is the maximum value among the three numbers. Note: You can create other static methods besides `Max3`. After that, call the `Max3` static method in your main method.

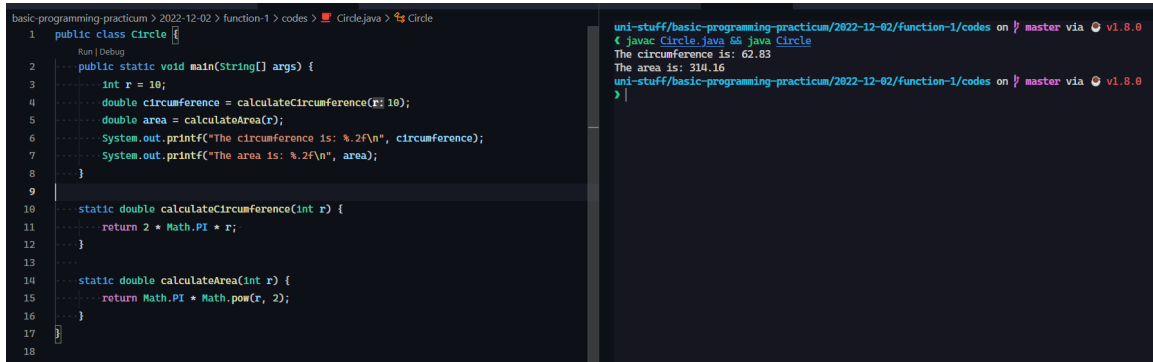


```
basic-programming-practicum > 2022-12-02 > function-1 > codes > AssignmentOne.java > Assign
1 public class AssignmentOne {
2     public static void main(String[] args) {
3         int maxValue = Max3(bil1: 2, bil2: 3, bil3: 4);
4         System.out.println("The max value is: " + maxValue);
5     }
6
7     static int Max3(int bil1, int bil2, int bil3) {
8         int maxValue = bil1;
9         maxValue = bil2 > maxValue ? bil2 : maxValue;
10        maxValue = bil3 > maxValue ? bil3 : maxValue;
11        return maxValue;
12    }
13 }
14

> javac AssignmentOne.java && java AssignmentOne
The max value is: 4
> |
```

Figure 10: Assignment 1 code and output

2. Create a class called `Circle` in which there is a function to calculate the circumference of a circle and the area of a circle.

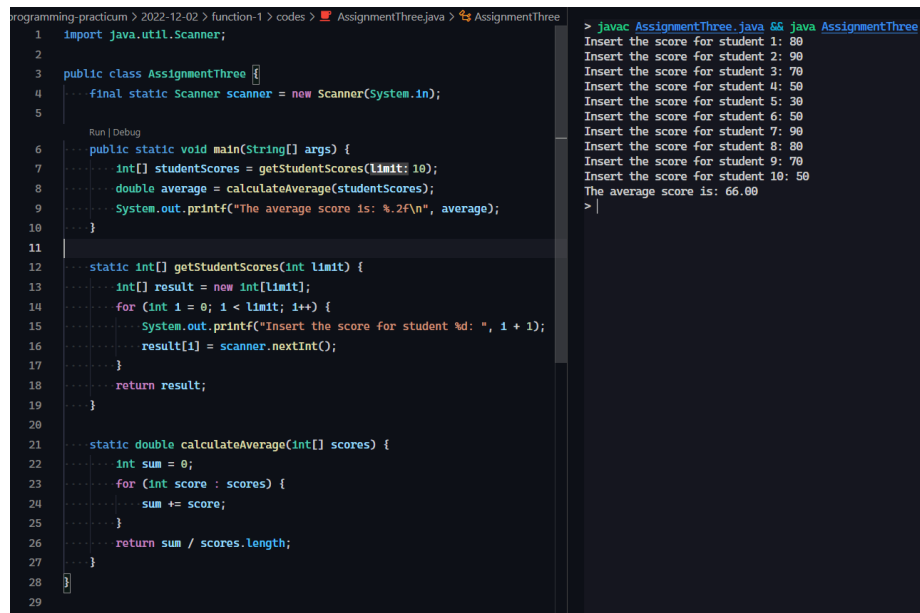


```
basic-programming-practicum > 2022-12-02 > function-1 > codes > Circle.java > Circle
1 public class Circle {
2     public static void main(String[] args) {
3         int r = 10;
4         double circumference = calculateCircumference(r);
5         double area = calculateArea(r);
6         System.out.printf("The circumference is: %.2f\n", circumference);
7         System.out.printf("The area is: %.2f\n", area);
8     }
9
10    static double calculateCircumference(int r) {
11        return 2 * Math.PI * r;
12    }
13
14    static double calculateArea(int r) {
15        return Math.PI * Math.pow(r, 2);
16    }
17 }
18

uni-stuff/basic-programming-practicum/2022-12-02/function-1/codes on / master via v1.8.0
< javac Circle.java && java Circle
The circumference is: 62.83
The area is: 314.16
uni-stuff/basic-programming-practicum/2022-12-02/function-1/codes on / master via v1.8.0
> |
```

Figure 11: Assignment 2 code and output

3. Create a program to fill array B with the data type `int` (10 students' test scores), where the input and filling process into the array is carried out in a function. Next, create another function to calculate the average value of the array (the average score of students tests). Print the average value, with the instructions for printing in the `main` function.



```
programming-practicum > 2022-12-02 > function-1 > codes > AssignmentThree.java > AssignmentThree
1 import java.util.Scanner;
2
3 public class AssignmentThree {
4     final static Scanner scanner = new Scanner(System.in);
5
6     public static void main(String[] args) {
7         int[] studentScores = getStudentScores(10);
8         double average = calculateAverage(studentScores);
9         System.out.printf("The average score is: %.2f\n", average);
10    }
11
12    static int[] getStudentScores(int limit) {
13        int[] result = new int[limit];
14        for (int i = 0; i < limit; i++) {
15            System.out.printf("Insert the score for student %d: ", i + 1);
16            result[i] = scanner.nextInt();
17        }
18        return result;
19    }
20
21    static double calculateAverage(int[] scores) {
22        int sum = 0;
23        for (int score : scores) {
24            sum += score;
25        }
26        return sum / scores.length;
27    }
28 }
29

> javac AssignmentThree.java && java AssignmentThree
Insert the score for student 1: 80
Insert the score for student 2: 90
Insert the score for student 3: 70
Insert the score for student 4: 50
Insert the score for student 5: 30
Insert the score for student 6: 60
Insert the score for student 7: 80
Insert the score for student 8: 80
Insert the score for student 9: 70
Insert the score for student 10: 50
The average score is: 66.00
> |
```

Figure 12: Assignment 3 code and output