JOBSHEET III
ARRAY OF OBJECTS

1.1 Learning Objective

At the end of this session, students will be able to:

1. Understand and explain the use of Array of Objects
2. Understand the logic of why we use Array of Objects in Java
3. Implement Array of Object in Java

1.2 Create, insert, and display Array of Object

In this session, we will have a practice of creating array of object, insert the data, and display it

1.2.1 Steps

1. Create a new project with name ArrayOfObjects. Create the package with name 'week3'
2. Create a **Rectangle** class:

```
public class Rectangle {
    public int length;
    public int width;
}
```

3. In main method in ArrayOfObjects class, create an array **Rectangle** and its length is 3

```
public class ArrayOfObjects {
    public static void main(String[] args) {
        Rectangle[] rectangleArray = new Rectangle[3];
    }
}
```

4. Then insert values for each the object's attributes

```
rectangleArray[0] = new Rectangle();
rectangleArray[0].length = 110;
rectangleArray[0].width = 30;

rectangleArray[1] = new Rectangle();
rectangleArray[1].length = 80;
rectangleArray[1].width = 40;

rectangleArray[2] = new Rectangle();
rectangleArray[2].length = 100;
rectangleArray[2].width = 20;
```

5. Print all the attributes object from ppArray as follows

```
System.out.println("First rectangle, width: " + rectangleArray[0].width + ", length: " + rectangleArray[0].length);
System.out.println("Second rectangle, width: " + rectangleArray[1].width + ", length: " + rectangleArray[1].length);
System.out.println("Third rectangle, width: " + rectangleArray[2].width + ", length: " + rectangleArray[2].length);
```

1.2.2 Result

Compile the code and see the result if it matches with following image.

```
run:
First rectangle, width: 30, length: 110
Second rectangle, width: 40, length: 80
Third rectangle, width: 20, length: 100
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.2.3 Questions

1. Based on practicum 1.2, does the class that are going to be used as an array of object must have attributes and methods? Please explain
2. Does class **Rectangle** have constructor? If not, why we instantiate the object as follows?

```
rectangleArray[1] = new Rectangle();
```

3. What's the meaning of this line of code?

```
Rectangle[] rectangleArray = new Rectangle[3];
```

4. What's the meaning of these lines of code?

```
rectangleArray[1] = new Rectangle();
rectangleArray[1].length = 80;
rectangleArray[1].width = 40;
```

5. Why **ArrayOfObject** class and **Rectangle** class should be differentiated?

1.3 Get input in Array of Objects using Loops

In this practicum we will update the program result in 1.2 so that the program could receive user inputs and use loops to assign values of each attributes of rectangles in ppArray

1.3.1 Steps

1. Import scanner in **ArrayOfObjects** class below the package declaration.

```
package ArrayOfObjects;
import java.util.Scanner;
```

2. In practicum 1.2 in 4th steps. Change the code as follows, this allows the Scanner object to be included in loops to receive input and assign user input values to the attributes.

```
        Rectangle[] rectangleArray = new Rectangle[3];
        Scanner sc = new Scanner(System.in);

        // Assign the values for each attribute in objects
        for (int i = 0; i < 3; i++) {
            rectangleArray[i] = new Rectangle();
            System.out.println("Rectangle " + i);

            System.out.print("Input length : ");
            rectangleArray[i].length = sc.nextInt();

            System.out.print("Input width : ");
            rectangleArray[i].width = sc.nextInt();;
        }
```

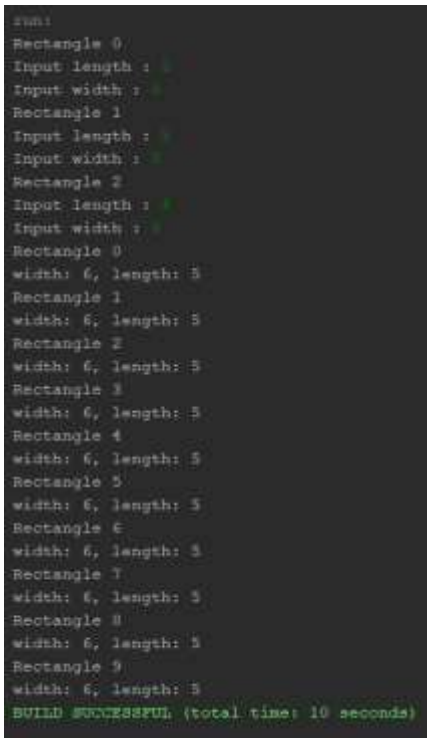
3. In practicum 1.2 in 5th steps. Change the code as follows. This time, we will use loop to access the element of **ppArray** and print it on the console

```
// Display the result in console
  for (int i = 0; i < 10; i++) {
      System.out.println("Rectangle " + i);
      System.out.println("width: " + rectangleArray[0].width + ", length: " + rectangleArray[0].length);
  }
```

4. See the result

1.3.2 Result

Run the program and see if it matches with following result:



1.3.3 Questions

1. Does array of object can be implemented on 2D array?
2. If yes, then please give an example. Otherwise, please explain?
3. There is a **Square** class that has an attribute **side** with integer as its data type. There will be an error when we run this code, why?

```
Square[] squareArray = new Square[100];
squareArray[5].side = 20;
```

4. Modify the code on practicum 1.3 so that the length of the array will be defined from user input
5. Can we duplicate the instantiation process in array of objects? For example, we assign the object in **ppArray[i]** and **ppArray[0]**, the instantiation process of **ppArray[0]** will be done twice. What's the effect of this?

1.4 Mathematical operation in array of object's attribute

1.4.1 Steps

1. Create a new package called **ArrayBlock**
2. Create a class named **Blocks**

```
public class Blocks {
    public int width, length, height;

    public Blocks(int p, int l, int t){
        length = p;
        width = l;
        height = t;
    }

    public int countVolume(){
        return length*width*height;
    }
}
```

3. In **main** function in **ArrayBlock,** instantiate array of **Blocks** that has size of 3

```
public class ArrayBlocks {
    public static void main(String[] args) {
        Blocks[] blArray = new Blocks[3];
    }
}
```

4. Then add these following codes to insert the value of **blArray** using its constructor

```
    blArray[0] = new Blocks(100, 30, 12);
    blArray[1] = new Blocks(120, 40, 15);
    blArray[2] = new Blocks(210, 50, 25);
```

5. Display the volume of all blocks by calling the method countVolume() in loop as follows.

```
for (int i = 0; i < 3; i++) {
    System.out.println("Volume blocks - " + i + " : " + blArray[i].countVolume());
}
```

6. Run and observe the result

1.4.2 Result

Run the program and see if it matches with following result:

```
run:
Volume blocks - 0 : 36000
Volume blocks - 1 : 72000
Volume blocks - 2 : 262500
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.4.3 Questions

1. Can we have more than one constructor in one class? Please explain
2. Create a **Triangle** class as follows

```
public class Triangle{
        public int base;
        public int height;
}
```

   Add another constructor in this class that has parameter **int a, int t**. These represents its base and height.

3. Add method **countArea()** and **countPerimeter()** in class **Triangle**
4. In main function, instantiate array of **Triangle** objects. Assign the attributes values as follows:

0th trArray	base: 10, height: 4
1st trArray	base: 20, height: 10
2nd trArray	base: 15, height: 6
3rd trArray	base: 25, height: 10

5. Display the result of area and perimeter for each triangle by calling the method **countArea()** and **countPerimeter()**

1.5 Practice

1. Create a program that can count surface area and volume of some 3D Geometry object (Cube, blocks, cylinder, etc). Then, create one more class to instantiate the array of objects with its constructor to assign values of its attributes.

   Note: Create loop to get user input and assign it to the attributes of the objects, then display the surface area and volume of each 3rd geometry object in console

2. A company that handles land transaction needs a program to calculate land area. This program must receive user input to assign values of these:
   - How many lands?
   - Length and width of the land

   This program calculates the area of inputted land information as its output. Check this following program:

```
How many lands: 3

Land 1
Length: 100
Width : 40

Land 2
Length: 250
```

```
Width : 100

Land 3
Length: 120
Width : 100

Land Area 1: 4000
Land Area 2: 25000
Land Area 3: 12000
```

3. Modify the program above so that it can display the widest area. (Additional note: create a different function to get the widest area)

```
Land 1
Length: 100
Width : 40

Land 2
Length: 250
Width : 100

Land 3
Length: 120
Width : 100

Land Area 1: 4000
Land Area 2: 25000
Land Area 3: 12000

The widest land is Land 2
```

4. A university needs a program to display student's information such as name, nim, gender, and GPA. This program should be able to receive input from all of those informations and display it to the user. Implement the program if there is 3 data sample, here is a reference of how you do it:

```
Insert 1st student data
Insert name :Rina
Insert nim :1234567
Insert gender :P
Insert IPK :3.5

Insert 2nd student data
Insert name :Rio
Insert nim :7654321
Insert gender:L
Insert IPK :4.0

Insert 3rd student data
Insert name :Reza
Insert nim :8765398
Insert gender:L
Insert IPK :3.8
```

Result:

```
1st Student Data
name : Rina
nim : 1234567
gender: P
IPK score: 3.5

2nd Student Data
name : Rio
nim : 7654321
gender: L
IPK score: 4.0

3rd student Data
```

```
name : Reza
nim : 8765398
gender: L
IPK score: 3.8
```

5. Modify the program above so that it can receive the average of IPK score from all students. (Note: create a new function to calculate the average of IPK Score in class **Students**)

```
Insert 1st student data
Insert name :Rina
Insert nim :1234567
Insert gender :P
Insert IPK :3.5

Insert 2nd student data
Insert name :Rio
Insert nim :7654321
Insert gender:L
Insert IPK :4.0

Insert 3rd student data
Insert name :Reza
Insert nim :8765398
Insert gender:L
Insert IPK :3.8
```

Result:

```
1st Student Data
name : Rina
nim : 1234567
gender: P
IPK score: 3.5

2nd Student Data
name : Rio
nim : 7654321
gender: L
IPK score: 4.0

3rd student Data
name : Reza
nim : 8765398
gender: L
IPK score: 3.8

Average IPK of all students : 3.7666667
```