JOBSHEET III
ARRAY OF OBJE CTS

3.1 Objectives of the Practicum

After completing this practical material, students are able to:

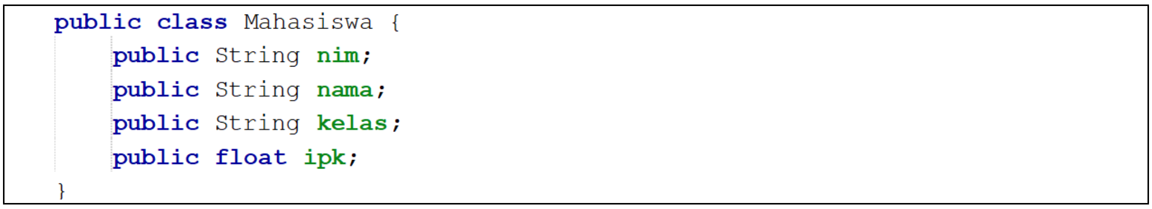
1. Understand and explain the function of arrays containing object variables.
2. Students are able to implement instantiation of arrays of objects in Java.
3. Students are able to perform operations on elements in an array of objects.

3.2 Creating an Array from Objects, Filling and Displaying

In this practical, we will practice how to create an array of objects, then fill and display the array.

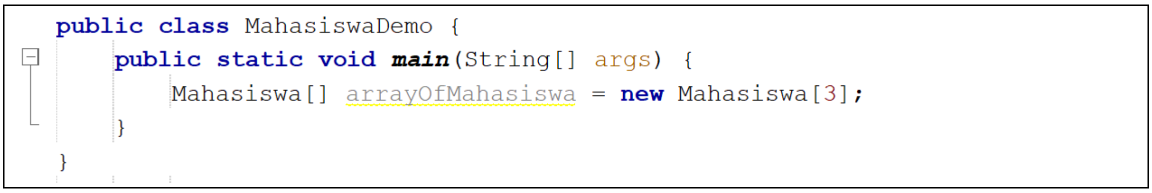
3.2.1 Experimental Steps

1. Create a new folder with the name Praktikum03 .
2. Create a class **`Student<`** `Presence No` **`>`** :



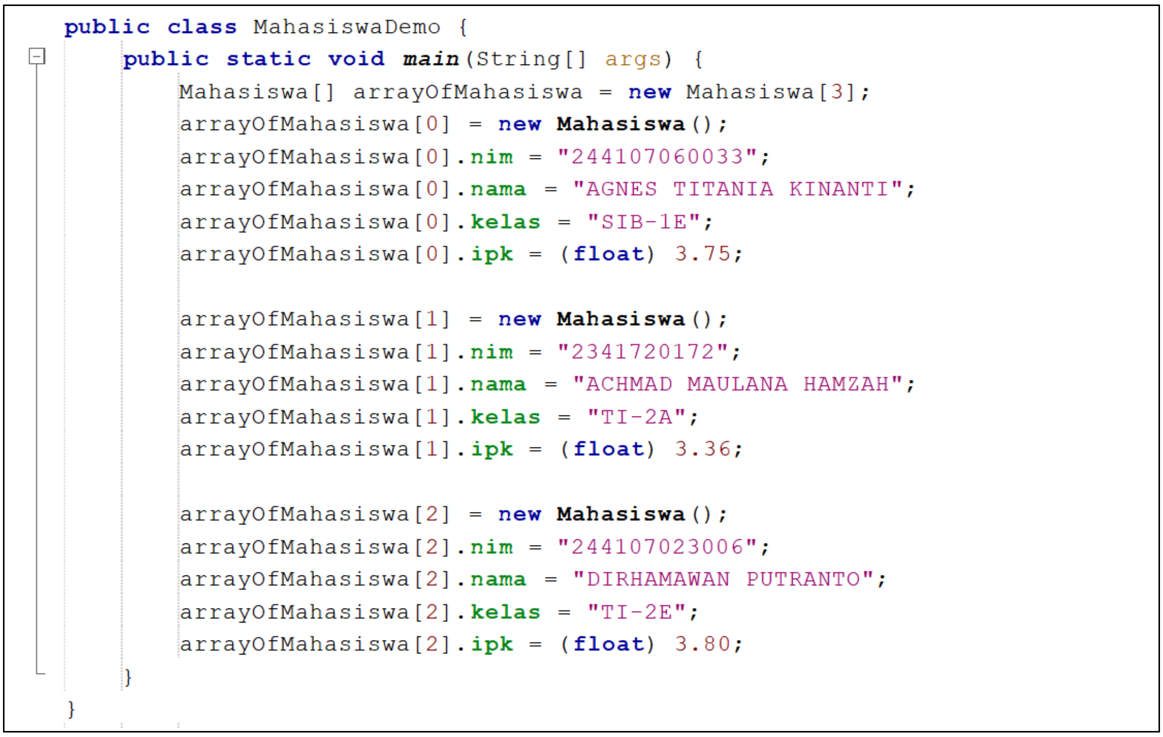
```
public class Mahasiswa {
    public String nim;
    public String nama;
    public String kelas;
    public float ipk;
}
```

3. Create a class **`MahasiswaDemo<NoPresensi>`** `then` add the main function as follows ;



```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
    }
}
```

4. Then fill in each attribute:

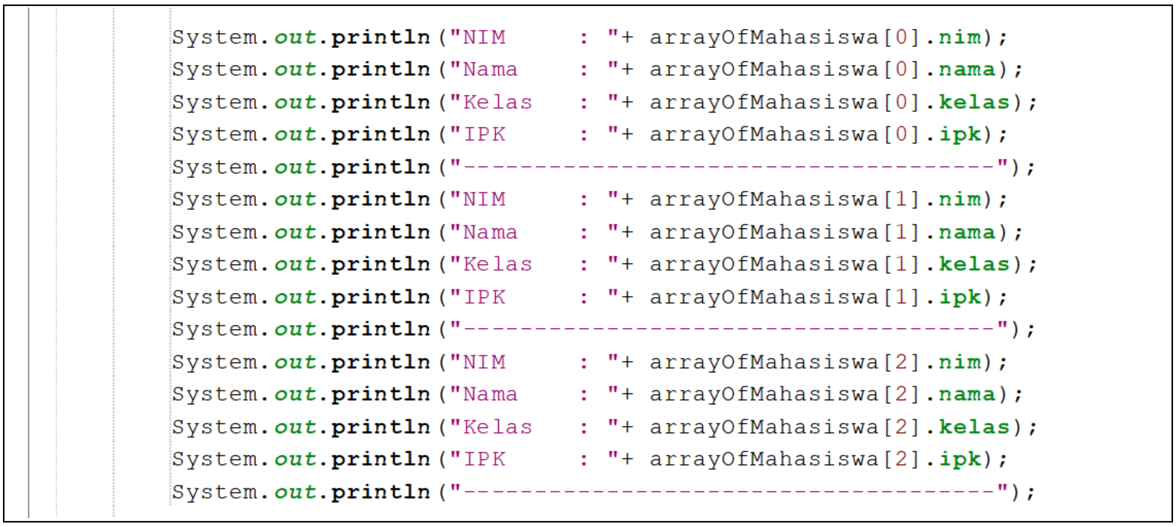


```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
        arrayOfMahasiswa[0] = new Mahasiswa();
        arrayOfMahasiswa[0].nim = "244107060033";
        arrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
        arrayOfMahasiswa[0].kelas = "SIB-1E";
        arrayOfMahasiswa[0].ipk = (float) 3.75;

        arrayOfMahasiswa[1] = new Mahasiswa();
        arrayOfMahasiswa[1].nim = "2341720172";
        arrayOfMahasiswa[1].nama = "ACHMAD MAULANA HAMZAH";
        arrayOfMahasiswa[1].kelas = "TI-2A";
        arrayOfMahasiswa[1].ipk = (float) 3.36;

        arrayOfMahasiswa[2] = new Mahasiswa();
        arrayOfMahasiswa[2].nim = "244107023006";
        arrayOfMahasiswa[2].nama = "DIRHAMAWAN PUTRANTO";
        arrayOfMahasiswa[2].kelas = "TI-2E";
        arrayOfMahasiswa[2].ipk = (float) 3.80;
    }
}
```

5. Print to the screen all attributes of the **`arrayOfStudents object`** :

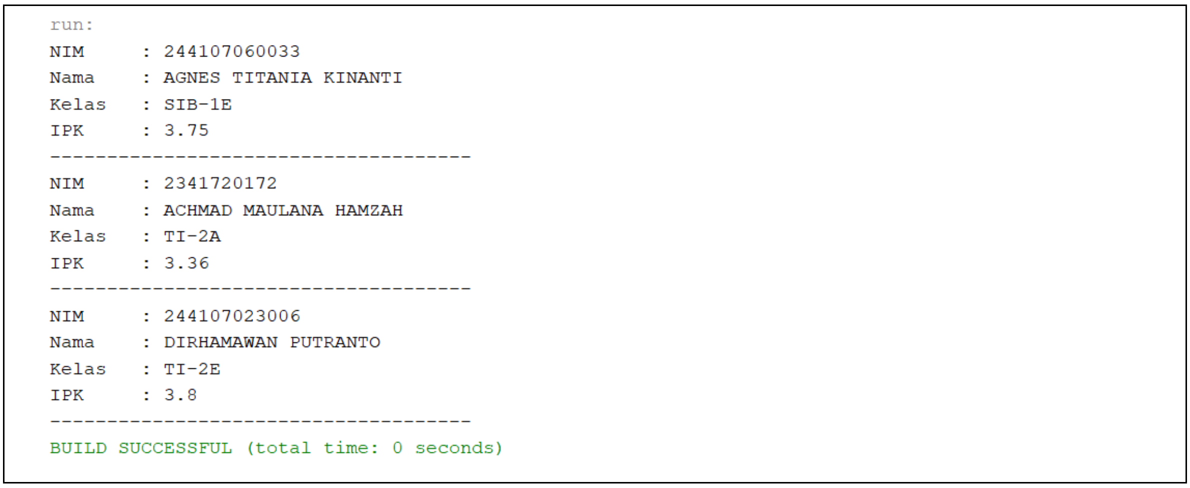


```
System.out.println("NIM     : "+ arrayOfMahasiswa[0].nim);
System.out.println("Nama    : "+ arrayOfMahasiswa[0].nama);
System.out.println("Kelas   : "+ arrayOfMahasiswa[0].kelas);
System.out.println("IPK     : "+ arrayOfMahasiswa[0].ipk);
System.out.println("---------------------------------------");
System.out.println("NIM     : "+ arrayOfMahasiswa[1].nim);
System.out.println("Nama    : "+ arrayOfMahasiswa[1].nama);
System.out.println("Kelas   : "+ arrayOfMahasiswa[1].kelas);
System.out.println("IPK     : "+ arrayOfMahasiswa[1].ipk);
System.out.println("---------------------------------------");
System.out.println("NIM     : "+ arrayOfMahasiswa[2].nim);
System.out.println("Nama    : "+ arrayOfMahasiswa[2].nama);
System.out.println("Kelas   : "+ arrayOfMahasiswa[2].kelas);
System.out.println("IPK     : "+ arrayOfMahasiswa[2].ipk);
System.out.println("---------------------------------------");
```

6. Run the program and observe the results.

3.2.2 Verification of Experimental Results

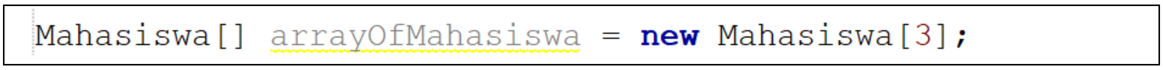
Match the results of compiling your program code with the following image.



```
run:
NIM     : 244107060033
Nama    : AGNES TITANIA KINANTI
Kelas   : SIB-1E
IPK     : 3.75
-------------------------------------
NIM     : 2341720172
Nama    : ACHMAD MAULANA HAMZAH
Kelas   : TI-2A
IPK     : 3.36
-------------------------------------
NIM     : 244107023006
Nama    : DIRHAMAWAN PUTRANTO
Kelas   : TI-2E
IPK     : 3.8
-------------------------------------
BUILD SUCCESSFUL (total time: 0 seconds)
```

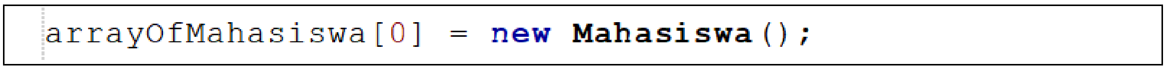
3.2.3 Question

1. Based on test 3.2, does **a class** that will create **an array of objects** always have to have both **attributes** and **methods** ? Explain!
2. What does the following program code do?



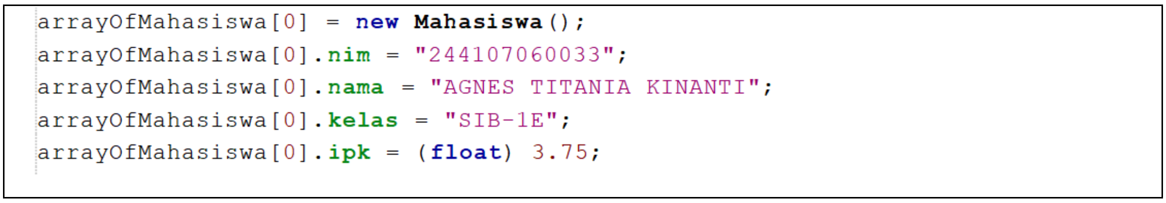
```
Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
```

3. Does the `Student class` have a constructor? If not, why is the constructor called in the following line of the program?



```
arrayOfMahasiswa[0] = new Mahasiswa();
```

4. What does the following program code do?



```
arrayOfMahasiswa[0] = new Mahasiswa();
arrayOfMahasiswa[0].nim = "244107060033";
arrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
arrayOfMahasiswa[0].kelas = "SIB-1E";
arrayOfMahasiswa[0].ipk = (float) 3.75;
```

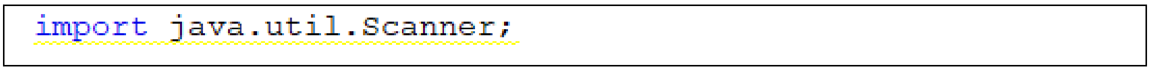
5. `the Student` and `StudentDemo` classes separated in test 3.2?

3.3 Receiving Array Input Using Looping

In this practical we will change the program results from practical 3.2 so that the program can accept input and use looping to fill in the attributes of all Student objects.

3.3.1 Experimental Steps

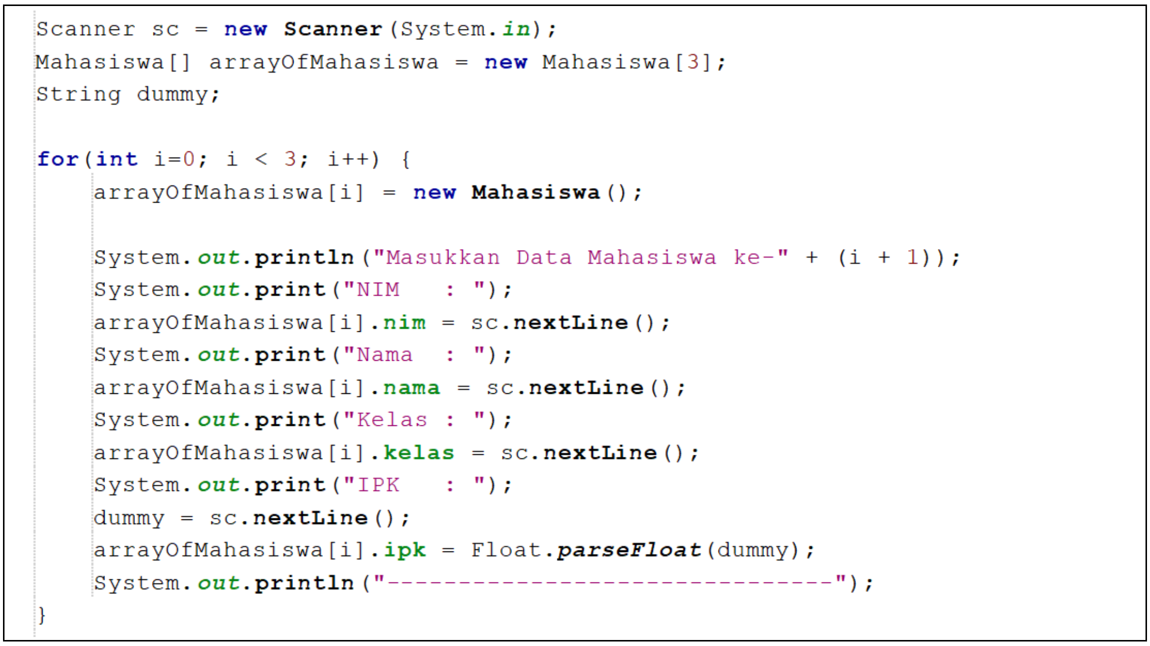
1. Import scanner in the **`MahasiswaDemo class`** .



```
import java.util.Scanner;
```

Note: Place the import code below the package code (if any).

2. In practical 3.2, point 4, modify the program code as follows. Create a **`Scanner object`** to receive input, then loop to retrieve the length and width information:

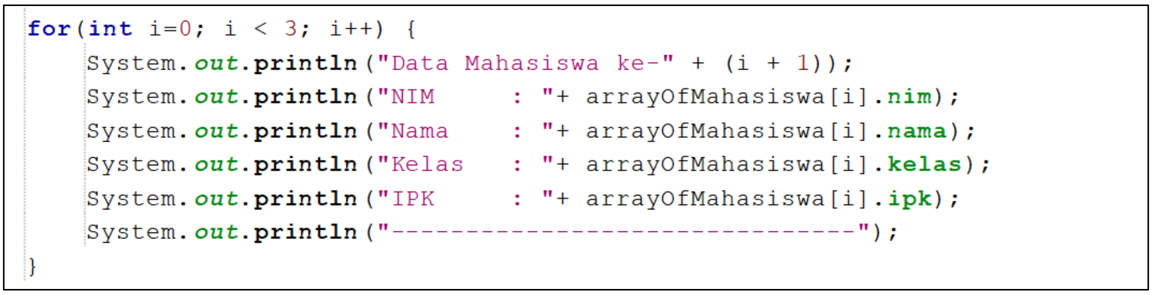


```
Scanner sc = new Scanner(System.in);
Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
String dummy;

for(int i=0; i < 3; i++) {
    arrayOfMahasiswa[i] = new Mahasiswa();

    System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));
    System.out.print("NIM   : ");
    arrayOfMahasiswa[i].nim = sc.nextLine();
    System.out.print("Nama  : ");
    arrayOfMahasiswa[i].nama = sc.nextLine();
    System.out.print("Kelas : ");
    arrayOfMahasiswa[i].kelas = sc.nextLine();
    System.out.print("IPK   : ");
    dummy = sc.nextLine();
    arrayOfMahasiswa[i].ipk = Float.parseFloat(dummy);
    System.out.println("--------------------------------");
}
```

3. In practical 3.2, point 5, modify the program code as follows. Loop through the array elements of **`the student objects`** and display their information on the screen:

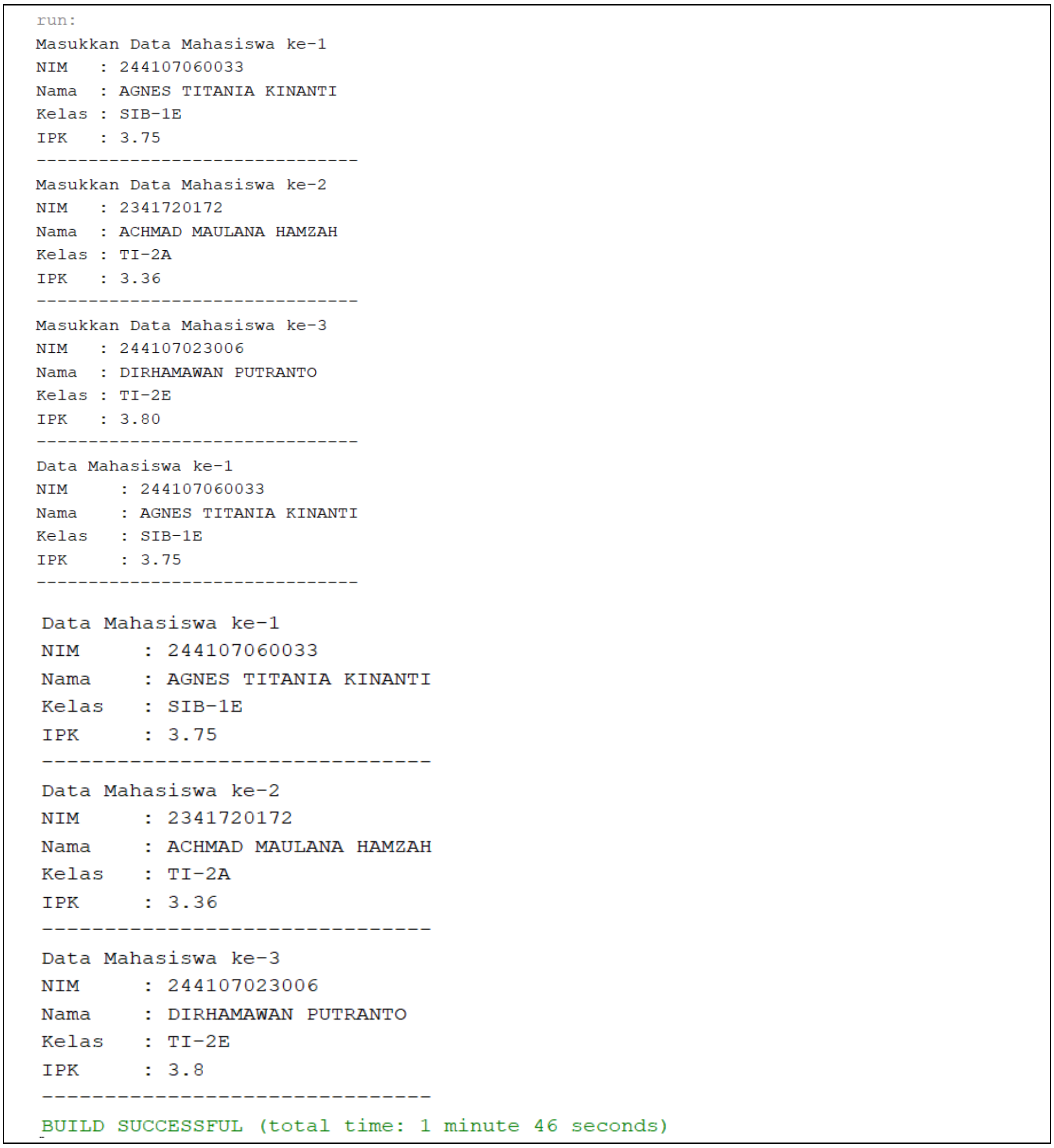


```
for(int i=0; i < 3; i++) {
    System.out.println("Data Mahasiswa ke-" + (i + 1));
    System.out.println("NIM      : "+ arrayOfMahasiswa[i].nim);
    System.out.println("Nama     : "+ arrayOfMahasiswa[i].nama);
    System.out.println("Kelas    : "+ arrayOfMahasiswa[i].kelas);
    System.out.println("IPK      : "+ arrayOfMahasiswa[i].ipk);
    System.out.println("--------------------------------");
}
```

4. Run the program and observe the results.

3.3.2 Verification of Experimental Results

Example of verification of the results of this experiment.

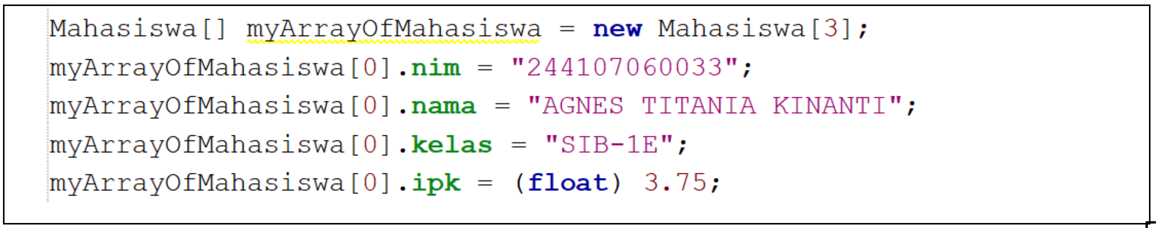


```
run:
Masukkan Data Mahasiswa ke-1
NIM   : 244107060033
Nama  : AGNES TITANIA KINANTI
Kelas : SIB-1E
IPK   : 3.75
-------------------------------
Masukkan Data Mahasiswa ke-2
NIM   : 2341720172
Nama  : ACHMAD MAULANA HAMZAH
Kelas : TI-2A
IPK   : 3.36
-------------------------------
Masukkan Data Mahasiswa ke-3
NIM   : 244107023006
Nama  : DIRHAMAWAN PUTRANTO
Kelas : TI-2E
IPK   : 3.80
-------------------------------
Data Mahasiswa ke-1
NIM     : 244107060033
Nama    : AGNES TITANIA KINANTI
Kelas   : SIB-1E
IPK     : 3.75
-------------------------------

Data Mahasiswa ke-1
NIM      : 244107060033
Nama     : AGNES TITANIA KINANTI
Kelas    : SIB-1E
IPK      : 3.75
--------------------------------
Data Mahasiswa ke-2
NIM      : 2341720172
Nama     : ACHMAD MAULANA HAMZAH
Kelas    : TI-2A
IPK      : 3.36
--------------------------------
Data Mahasiswa ke-3
NIM      : 244107023006
Nama     : DIRHAMAWAN PUTRANTO
Kelas    : TI-2E
IPK      : 3.8
--------------------------------
BUILD SUCCESSFUL (total time: 1 minute 46 seconds)
```

3.3.3 Question

1. Add **the printInfo() method** to the **Student class** then modify the program code in step no. 3.
2. Suppose you have a new array **array of** type **Students** named **myArrayOfStudents** . Why does the following code cause an error?



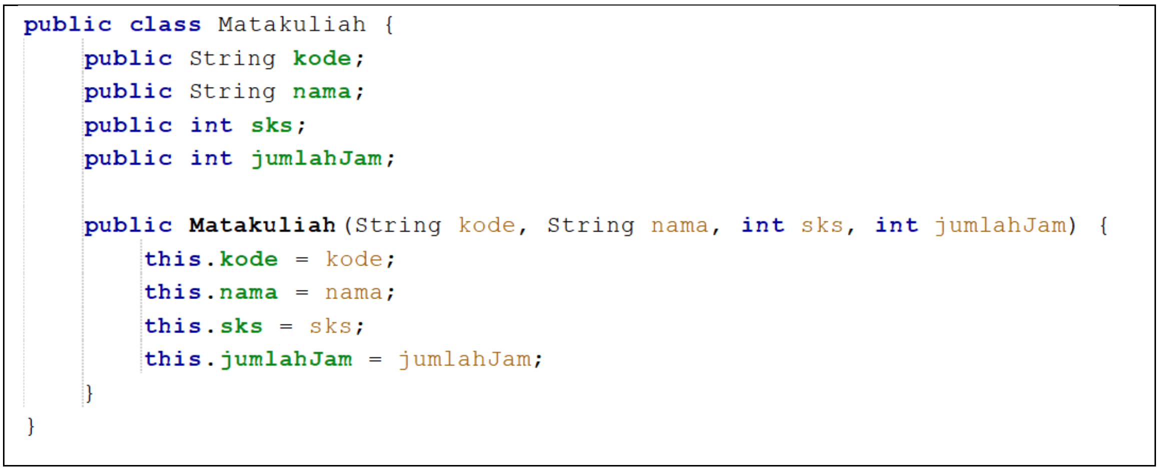
```
Mahasiswa[] myArrayOfMahasiswa = new Mahasiswa[3];
myArrayOfMahasiswa[0].nim = "244107060033";
myArrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
myArrayOfMahasiswa[0].kelas = "SIB-1E";
myArrayOfMahasiswa[0].ipk = (float) 3.75;
```

3.4 Parameterized Constructor

In this practical, we will instantiate an `array of objects` using a parameterized `constructor` .

3.4.1 Experimental Steps

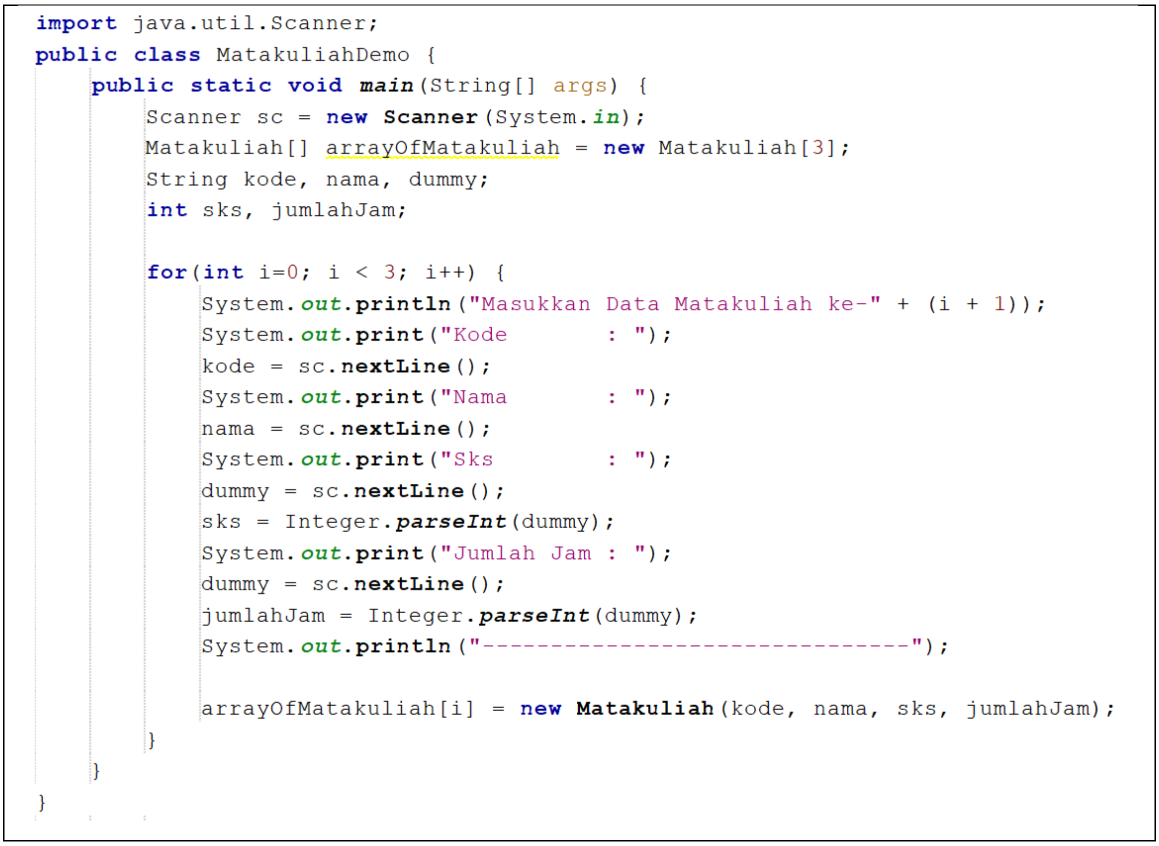
1. **`a`** new class with the name **`Matakuliah<NoPresensi>`** `with` a **`constructor`** with the following parameters;



```
public class Matakuliah {
    public String kode;
    public String nama;
    public int sks;
    public int jumlahJam;

    public Matakuliah(String kode, String nama, int sks, int jumlahJam) {
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
        this.jumlahJam = jumlahJam;
    }
}
```

2. **`a`** new class **`MatakuliahDemo<NoPresensi>`** `and` add the main() function. Then, instantiate **`the Matakuliah object array`** using the following parameterized **`constructor`** **`;`**



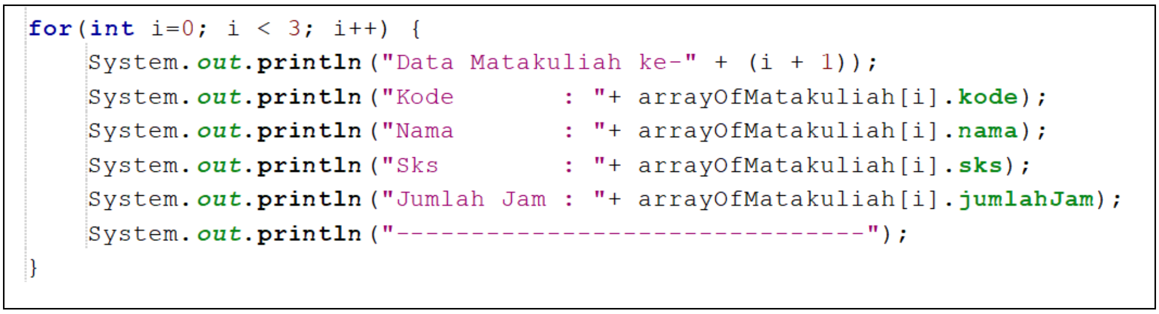
```
import java.util.Scanner;
public class MatakuliahDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Matakuliah[] arrayOfMatakuliah = new Matakuliah[3];
        String kode, nama, dummy;
        int sks, jumlahJam;

        for(int i=0; i < 3; i++) {
            System.out.println("Masukkan Data Matakuliah ke-" + (i + 1));
            System.out.print("Kode       : ");
            kode = sc.nextLine();
            System.out.print("Nama       : ");
            nama = sc.nextLine();
            System.out.print("Sks        : ");
            dummy = sc.nextLine();
            sks = Integer.parseInt(dummy);
            System.out.print("Jumlah Jam : ");
            dummy = sc.nextLine();
            jumlahJam = Integer.parseInt(dummy);
            System.out.println("-------------------------------");

            arrayOfMatakuliah[i] = new Matakuliah(kode, nama, sks, jumlahJam);
        }
    }
}
```

3. Run the program then observe the results.

4. Modify the **`MatakuliahDemo class`** so that it can display the input results of the variable **`array of Matakuliah objects`** in the layer;

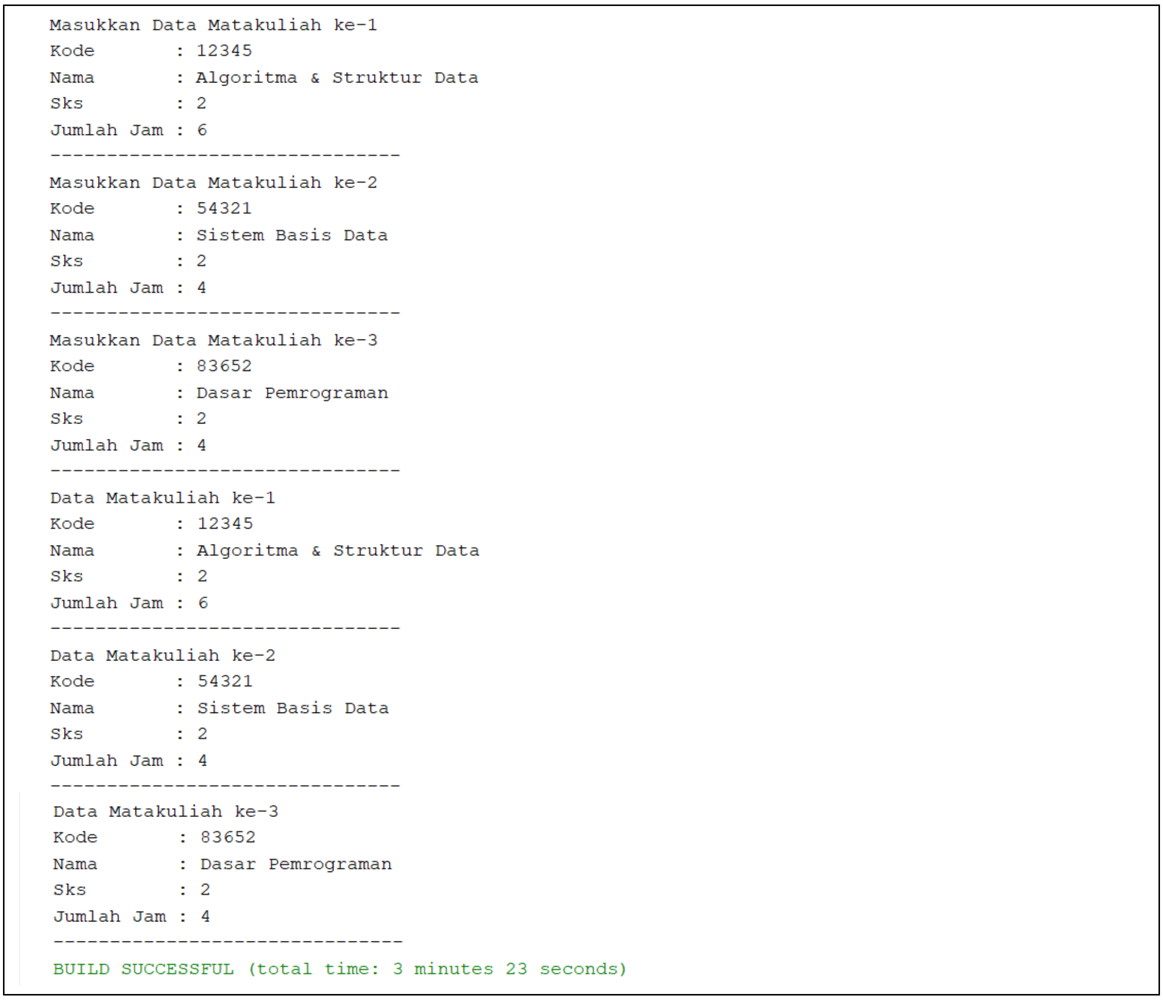


```
for(int i=0; i < 3; i++) {
    System.out.println("Data Matakuliah ke-" + (i + 1));
    System.out.println("Kode        : "+ arrayOfMatakuliah[i].kode);
    System.out.println("Nama        : "+ arrayOfMatakuliah[i].nama);
    System.out.println("Sks         : "+ arrayOfMatakuliah[i].sks);
    System.out.println("Jumlah Jam : "+ arrayOfMatakuliah[i].jumlahJam);
    System.out.println("-------------------------------");
}
```

5. Run the program and observe the results.

3.4.2 Verification of Experimental Results

Match the results of compiling your program code with the following image.



```
Masukkan Data Matakuliah ke-1
Kode        : 12345
Nama        : Algoritma & Struktur Data
Sks         : 2
Jumlah Jam : 6
-------------------------------
Masukkan Data Matakuliah ke-2
Kode        : 54321
Nama        : Sistem Basis Data
Sks         : 2
Jumlah Jam : 4
-------------------------------
Masukkan Data Matakuliah ke-3
Kode        : 83652
Nama        : Dasar Pemrograman
Sks         : 2
Jumlah Jam : 4
-------------------------------
Data Matakuliah ke-1
Kode        : 12345
Nama        : Algoritma & Struktur Data
Sks         : 2
Jumlah Jam : 6
-------------------------------
Data Matakuliah ke-2
Kode        : 54321
Nama        : Sistem Basis Data
Sks         : 2
Jumlah Jam : 4
-------------------------------
Data Matakuliah ke-3
Kode        : 83652
Nama        : Dasar Pemrograman
Sks         : 2
Jumlah Jam : 4
-------------------------------
BUILD SUCCESSFUL (total time: 3 minutes 23 seconds)
```

3.4.3 Question

1. Can a class have more than 1 constructor? If yes, give an example.
2. Add the **`addData()`** **`method`** to **`the Matakuliah class`** , then use this method in **`the MatakuliahDemo class`** to add Course data.
3. Add **`the printInfo() method`** to **`the Matakuliah class`** , then use this method in **`the MatakuliahDemo class`** to display the input data on the screen.
4. Modify the program code in the **`MatakuliahDemo class`** so that the length (number of elements) of **`the array of Matakuliah objects`** is determined by the user via input with Scanner

3.5 Task

1. Create a program to display information about lecturers. The program can accept input of all lecturer-related information and display it back on the screen. This program consists of the Dosen<NoPresensi> class **`with the`** `following` attributes/properties:

   Code string
   Name string
   Boolean **gender**
   int **age**

   with the constructor method as follows;

   public lecturer(String **code,** String **name,** Boolean **gender,** int **age**) {
   …………….;
   …………….;
   }

   , create a class **`DosenDemo<NoPresensi>`** `to` process input and display data for several lecturers. Use a loop with **`FOR`** To create **`an array of objects , use FOREACH`** looping. to display data on the screen.

2. Add a new class **`DataDosen<NoPresensi>`** `with` the following methods `;`

   a. **`dataAllDosen(Dosen[] arrayOfDosen)`** to display data for all lecturers

   b. **`numberOfLecturersPerGender(Lecturers[] arrayOfLecturers)`** to display data on the number of lecturers per gender (Male / Female)

   c. **`AverageAgeOfLecturersPerGender`** (**`Lecturers[] arrayOfLecturers`**) to display the average age of lecturers per gender (Male / Female)

   d. **`infoDosenPalingTua`** (**`Dosen[] arrayOfDosen`**) to display the oldest lecturer data

   e. **`infoDosenMostYoungest`** (**`Dosen[] arrayOfDosen`**) to display the youngest lecturer data

   All these methods must be able to be called/tested from the **`DosenDemo class.`**