

# Basic Programming Practicum

## Jobsheet 8 Array 1



**Name**

Dicha Zelianivan Arkana

**NIM**

2241720002

**Class**

1i

**Department**

Information Technology

**Study Program**

D4 Informatics Engineering

---

# 1 Laboratory

## 1.1 Experiment 1: Fill in Array Element

1. Create a new project
2. Create a new class, name it **myArray**
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Create an array of integer type named **num** with a capacity of 4 elements

```
int[] num = new int[4];
```

5. Fill each element of the array with numbers 5, 12, 7, 20

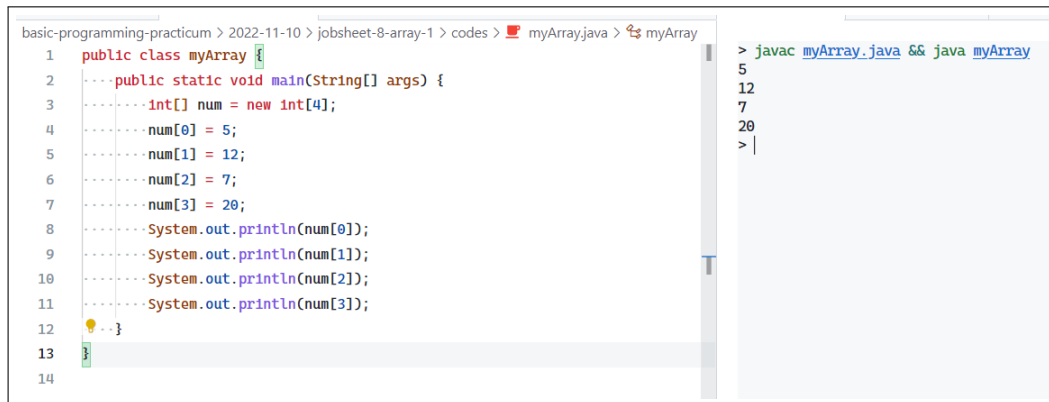
```
num[0] = 5;  
num[1] = 12;  
num[2] = 7;  
num[3] = 20;
```

6. Display all contents of the elements to the screen

```
System.out.println(num[0]);  
System.out.println(num[1]);  
System.out.println(num[2]);  
System.out.println(num[3]);
```

- 
7. Compile and run the program. Match the result of the running programs that you have created according to the following display

5  
12  
7  
20



The screenshot shows an IDE with two panes. The left pane displays the source code for `myArray.java`:

```
1 public class myArray {  
2     public static void main(String[] args) {  
3         int[] num = new int[4];  
4         num[0] = 5;  
5         num[1] = 12;  
6         num[2] = 7;  
7         num[3] = 20;  
8         System.out.println(num[0]);  
9         System.out.println(num[1]);  
10        System.out.println(num[2]);  
11        System.out.println(num[3]);  
12    }  
13  
14 }
```

The right pane shows the terminal output after running the command `javac myArray.java && java myArray`:

```
> javac myArray.java && java myArray  
5  
12  
7  
20  
> |
```

Figure 1: Experiment 1 code and output

## Questions

1. In Experiment 1, what are the largest and smaller array indexes?  
The smallest array index is 0 and the largest is 3
2. If the contents of each element of the array `num` are changed with numbers 5.0, 12867, 7.5, 2000000. What happens? How can it be like that?

The fractions will be truncated because we used the `int` type for our array which couldn't store fractions. Every elements in an array must have the same data type.

3. Change the statement in step 6 to be like this

```
for (int i = 0; i < 4; i++) {  
    System.out.println(num[i]);  
}
```

What is the result? How can it be like that?

The result is still the same because we used the looping construct instead of copying and pasting the lines manually. They both do the same thing, which is doing `System.out.println()` 4 times.

---

## 1.2 Experiment 2: Requesting User Input to Fill in an Array Element

1. Create a new class, name it **arrayInputLoop**
2. Write the basic structure of the Java programming language which contains the **main()** function
3. Add the Scanner library
4. Make a **Scanner** declaration with the name **sc**
5. Create an array of integer type with the name **finalScore**, with a capacity of 6 elements

```
int[] finalScore = new int[6];
```

6. Using a loop, create an input to fill in the **finalScore** array element

```
for (int i = 0; i < 6; i++) {  
    System.out.print("Enter the final score " + i + ": ");  
    finalScore[i] = sc.nextInt();  
}
```

7. Using a loop, display all the contents of the elements from the **finalScore** array

```
for (int i = 0; i < 6; i++) {  
    System.out.println("Final score " + i + " is " + finalScore[i]);  
}
```

- 
8. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
Enter the final score 0: 88
Enter the final score 1: 90
Enter the final score 2: 74
Enter the final score 3: 83
Enter the final score 4: 92
Enter the final score 5: 77
Final score 0 is 88
Final score 1 is 90
Final score 2 is 74
Final score 3 is 83
Final score 4 is 92
Final score 5 is 77
```



The screenshot shows an IDE with two panes. The left pane displays the source code for `arrayInputLoop.java`. The code imports `java.util.Scanner`, defines a `main` method, creates a `Scanner` object, and uses a `for` loop to read six integers into an array. The right pane shows the command prompt output, which matches the expected results from the previous block. The command `> javac arrayInputLoop.java && java arrayInputLoop` is entered, followed by the program's execution output.

```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > arrayInputLoop.java > ...
1  import java.util.Scanner;
2
3  public class arrayInputLoop {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int[] finalScore = new int[6];
7
8          for (int i = 0; i < 6; i++) {
9              System.out.print("Enter the final score " + i + ": ");
10             finalScore[i] = sc.nextInt();
11         }
12
13         for (int i = 0; i < 6; i++) {
14             System.out.println("Final score " + i + " is " + finalScore[i]);
15         }
16
17         sc.close();
18     }
19 }
20
> javac arrayInputLoop.java && java arrayInputLoop
Enter the final score 0: 88
Enter the final score 1: 90
Enter the final score 2: 74
Enter the final score 3: 83
Enter the final score 4: 92
Enter the final score 5: 77
Final score 0 is 88
Final score 1 is 90
Final score 2 is 74
Final score 3 is 83
Final score 4 is 92
Final score 5 is 77
>
```

Figure 2: Experiment 2 code and output

## Questions!

1. Change the statement in step 6 to be like this

```
for (int i = 0; i < finalScore.length; i++) {
    System.out.println("Enter the final score " + i + ": ");
    finalScore[i] = sc.nextInt();
}
```

Run the program. Have there been any changes? How can it be like that?

---

There is no changes. The reason is because instead of using a hardcoded value 6, we now use the property `.length` from the `finalScore` array.

2. What is the use of `finalScore.length`?

It is used to get the length of the `finalScore` array.

3. Change the statement in step 7 to be like this, so that the program only displays the grades of students who passed

```
for (int i = 0; i < finalScore.length; i++) {  
    if (finalScore[i] > 70) {  
        System.out.println("Final score " + i + " is " + finalScore[i]);  
    }  
}
```

Run the program and describe the flow of the program!

### Program Flow

- It initialises the `for` loop
- It checks if the current score is higher than 70
- If the current score is higher than 70, then print the score



The screenshot shows an IDE with two panels. The left panel displays the source code for `arrayInputLoop.java`. The code imports `java.util.Scanner`, defines a `main` method, creates a `Scanner` object, and initializes an `int` array `finalScore` of size 6. It then uses a `for` loop to prompt the user for 6 scores. Inside the loop, it checks if the score is greater than 70 and prints the score if it is. The right panel shows the command prompt output, where the user has entered scores 82, 78, 65, 88, 70, and 90. The program has printed 'Final score 0 is 82', 'Final score 1 is 78', 'Final score 3 is 88', and 'Final score 5 is 90'.

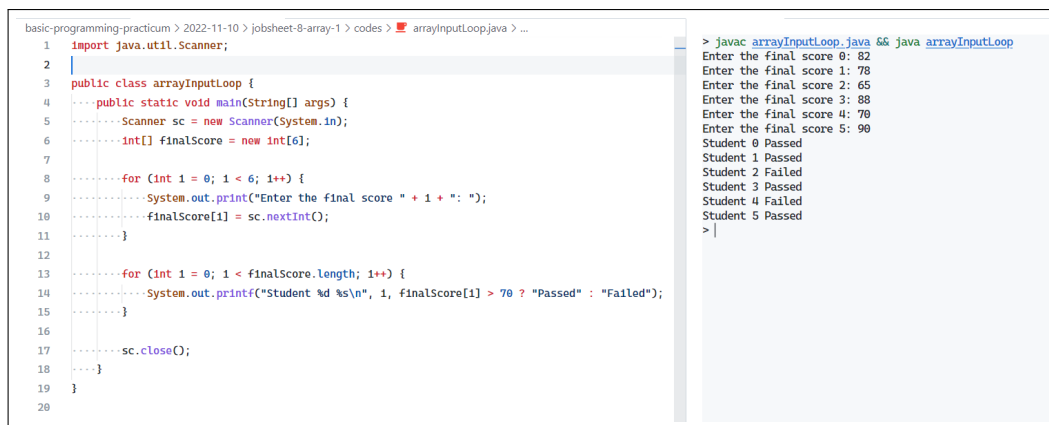
```
-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > arrayInputLoop.java > arrayInputLoop > main(String[])  
1  import java.util.Scanner;  
2  
3  public class arrayInputLoop {  
4  .....public static void main(String[] args) {  
5  .....Scanner sc = new Scanner(System.in);  
6  .....int[] finalScore = new int[6];  
7  
8  .....for (int i = 0; i < 6; i++) {  
9  .....System.out.print("Enter the final score " + i + ": ");  
10 .....finalScore[i] = sc.nextInt();  
11 .....}  
12  
13 .....for (int i = 0; i < finalScore.length; i++) {  
14 .....if (finalScore[i] > 70) {  
15 .....System.out.println("Final score " + i + " is " + finalScore[i]);  
16 .....}  
17 .....}  
18  
19 .....sc.close();  
20 .....}  
21 .....}  
--  
  
> javac arrayInputLoop.java && java arrayInputLoop  
Enter the final score 0: 82  
Enter the final score 1: 78  
Enter the final score 2: 65  
Enter the final score 3: 88  
Enter the final score 4: 70  
Enter the final score 5: 90  
Final score 0 is 82  
Final score 1 is 78  
Final score 3 is 88  
Final score 5 is 90  
> |
```

Figure 3: Experiment 2 with `>70` scores

4. Modify the program so that it displays all students, and marked which one passed and which did not!

---

```
Enter the final score 0: 82
Enter the final score 1: 78
Enter the final score 2: 65
Enter the final score 3: 88
Enter the final score 4: 70
Enter the final score 5: 90
Student 0 Passed
Student 1 Passed
Student 2 Failed
Student 3 Passed
Student 4 Failed
Student 5 Passed
```



The screenshot shows a code editor with two panes. The left pane displays the source code for `arrayInputLoop.java`, which imports `java.util.Scanner`, defines a `main` method, and uses a `Scanner` to read six scores. It then iterates through the scores, printing "Passed" or "Failed" based on whether the score is greater than 70. The right pane shows the terminal output of the program, which matches the text shown in the first block of the document.

```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > arrayInputLoop.java > ...
1  import java.util.Scanner;
2
3  public class arrayInputLoop {
4      ... public static void main(String[] args) {
5          ... Scanner sc = new Scanner(System.in);
6          ... int[] finalScore = new int[6];
7
8          ... for (int i = 0; i < 6; i++) {
9              ... System.out.print("Enter the final score " + i + ": ");
10             ... finalScore[i] = sc.nextInt();
11             ... }
12
13             ... for (int i = 0; i < finalScore.length; i++) {
14                 ... System.out.printf("Student %d %s\n", i, finalScore[i] > 70 ? "Passed" : "Failed");
15                 ... }
16
17             ... sc.close();
18             ... }
19     }
20
> javac arrayInputLoop.java && java arrayInputLoop
Enter the final score 0: 82
Enter the final score 1: 78
Enter the final score 2: 65
Enter the final score 3: 88
Enter the final score 4: 70
Enter the final score 5: 90
Student 0 Passed
Student 1 Passed
Student 2 Failed
Student 3 Passed
Student 4 Failed
Student 5 Passed
> |
```

Figure 4: Experiment 2 with Passed or Failed text

### 1.3 Experiment 3: Perform Arithmetic Operations on Array Elements

1. This experiment is done to add array elements. The program will accept input of 10 student scores. Then the program will display the average score of 10 students.
2. Create a new class, name it `averageScore`
3. Write the basic structure of the Java programming language which contains the `main()` function
4. Add the `Scanner` library
5. Make a **Scanner** declaration with the name `sc`

- 
6. Create an array of integer type with the name `score` with a capacity of 10. Then declare the variables `total` and `average`

```
int[] score = new int[10];
double total = 0;
double average;
```

7. Using a loop, create an input to fill in the `score` array element

```
for (int i = 0; i < score.length; i++) {
    System.out.print("enter student score " + (i + 1) + ": ");
    score[i] = sc.nextInt();
}
```

8. Using a loop, calculate the total number of scores.

```
for (int i = 0; i < score.length; i++) {
    total += score[i];
}
```

9. Calculate the average value by dividing `total` by the number of elements of `score`

```
average = total / score.length;
System.out.println("The class average score is " + average);
```



- 
10. Compile and run the program. Match the results of the running programs that you have created according to the following display

```
Enter student score 1: 98
Enter student score 2: 73
Enter student score 3: 86
Enter student score 4: 82
Enter student score 5: 95
Enter student score 6: 68
Enter student score 7: 90
Enter student score 8: 71
Enter student score 9: 78
Enter student score 10: 84
The class average score is 82.5
```



```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > averageScore.java > ...
1  import java.util.Scanner;
2
3  public class averageScore {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          int[] score = new int[10];
8          double total = 0;
9          double average;
10
11         for (int i = 0; i < score.length; i++) {
12             System.out.print("enter student score " + (i + 1) + ": ");
13             score[i] = sc.nextInt();
14         }
15
16         for (int i = 0; i < score.length; i++) {
17             total += score[i];
18         }
19
20         average = total / score.length;
21         System.out.println("The class average score is " + average);
22
23         sc.close();
24     }
25 }
26
```

```
> javac averageScore.java && java averageScore
enter student score 1: 98
enter student score 2: 73
enter student score 3: 86
enter student score 4: 82
enter student score 5: 95
enter student score 6: 68
enter student score 7: 90
enter student score 8: 71
enter student score 9: 78
enter student score 10: 84
The class average score is 82.5
> |
```

Figure 5: Experiment 3 code and output

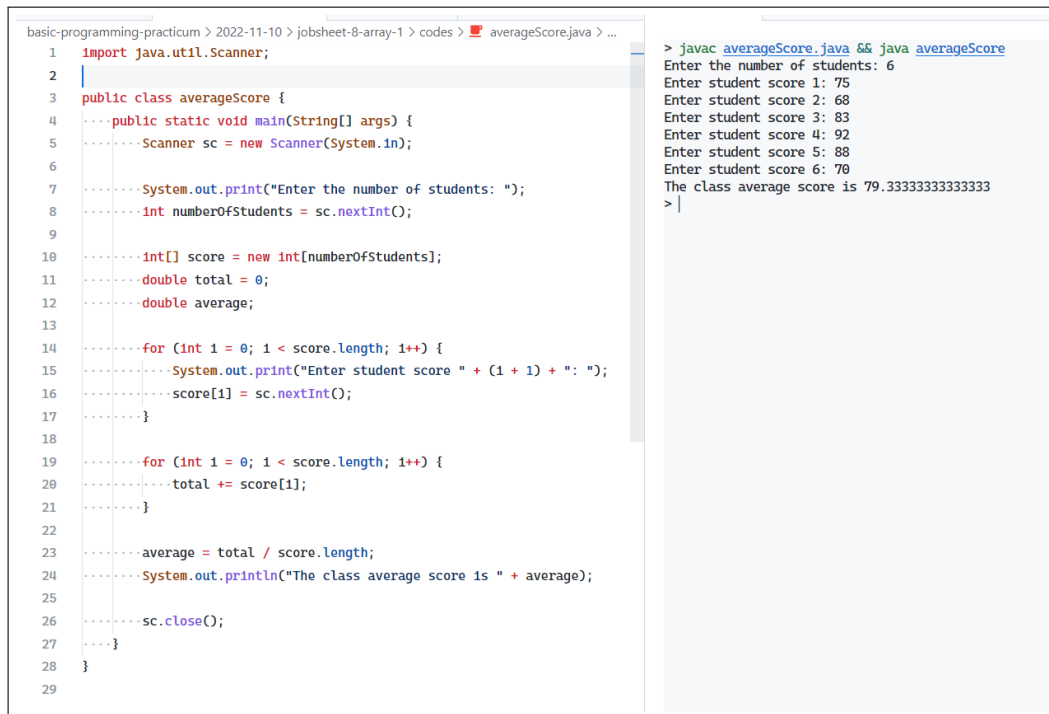
## Questions!

1. In step 9, why is the average calculation written outside the loop?

Because we want to calculate the average once and we need to wait for all of the scores to be accumulated into the `total` variable.

- 
2. Modify the program in Experiment 3 so that it can produce output like the following display

```
Enter the number of students: 6
Enter student score 1: 75
Enter student score 2: 68
Enter student score 3: 83
Enter student score 4: 92
Enter student score 5: 88
Enter student score 6: 70
The class average score is 79.3333333333
```



```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > averageScore.java > ...
1  import java.util.Scanner;
2
3  public class averageScore {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          System.out.print("Enter the number of students: ");
8          int numberOfStudents = sc.nextInt();
9
10         int[] score = new int[numberOfStudents];
11         double total = 0;
12         double average;
13
14         for (int i = 0; i < score.length; i++) {
15             System.out.print("Enter student score " + (i + 1) + ": ");
16             score[i] = sc.nextInt();
17         }
18
19         for (int i = 0; i < score.length; i++) {
20             total += score[i];
21         }
22
23         average = total / score.length;
24         System.out.println("The class average score is " + average);
25
26         sc.close();
27     }
28 }
29
```

```
> javac averageScore.java && java averageScore
Enter the number of students: 6
Enter student score 1: 75
Enter student score 2: 68
Enter student score 3: 83
Enter student score 4: 92
Enter student score 5: 88
Enter student score 6: 70
The class average score is 79.3333333333
> |
```

Figure 6: Experiment 3 with number of students input

---

## 2 Assignment

1. Create a program that has an array of 5 elements. Then use the input to fill in the array elements, and display the contents of the array in reverse order as in the following illustration (figure 7).

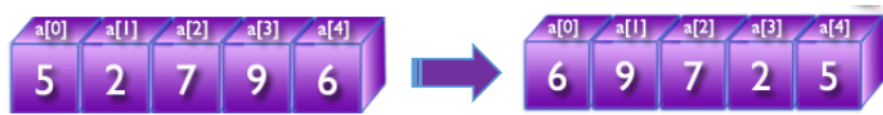


Figure 7: Array Illustration

```
> 2022-11-10 > jobsheet-8-array-1 > codes > AssignmentOne.java > AssignmentOne > main(String[])
1 public class AssignmentOne {
2     ... public static void main(String[] args) {
3         ... int[] numbers = { 5, 2, 7, 9, 6 };
4
5         ... for (int i = numbers.length - 1; i >= 0; i--) {
6             ... System.out.printf("%d ", numbers[i]);
7         }
8     }
9 }
10
```

```
> javac AssignmentOne.java && java AssignmentOne
6 9 7 2 5
> |
```

Figure 8: Assignment 1 code and output

- 
2. Create a program that accepts the number of array elements as input, also input the elements of array. Then display the largest number of the array elements. Examples of program results are as follows:

```
Enter the number of array elements: 4
Enter the value of element 1: 27
Enter the value of element 2: 8
Enter the value of element 3: 33
Enter the value of element 4: 11
The largest number is 33
```



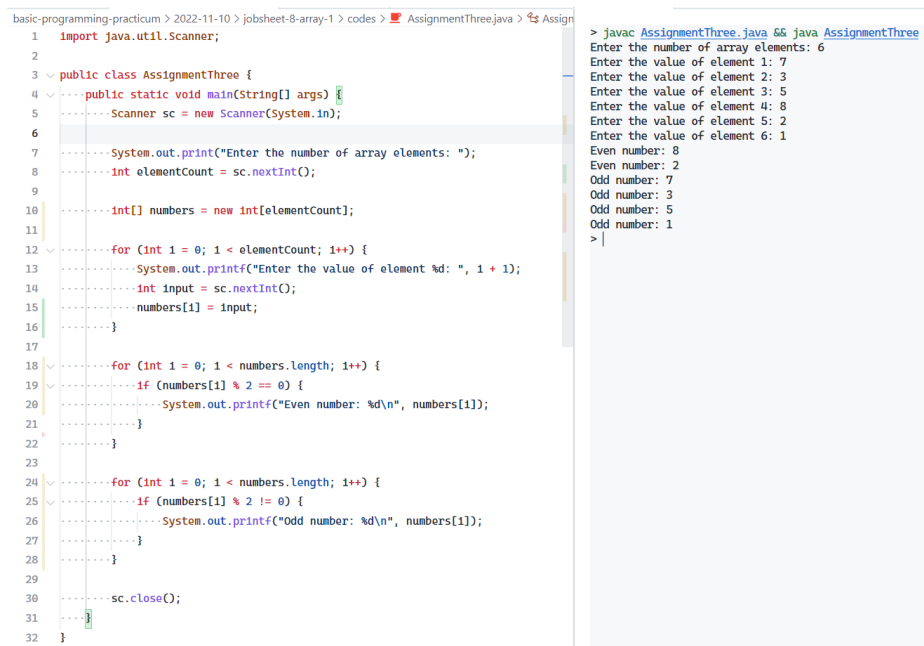
The screenshot displays a code editor with two panes. The left pane shows the source code for `AssignmentTwo.java`, which uses a `Scanner` to read the number of elements and their values, then finds the maximum. The right pane shows the terminal output of running the program, which matches the example results provided in the text.

```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > AssignmentTwo.java > ...
1  import java.util.Scanner;
2
3  public class AssignmentTwo {
4      ...public static void main(String[] args) {
5          ...Scanner sc = new Scanner(System.in);
6
7          ...System.out.print("Enter the number of array elements: ");
8          ...int elementCount = sc.nextInt();
9          ...int[] elements = new int[elementCount];
10
11         ...for (int i = 0; i < elements.length; i++) {
12             ...System.out.printf("Enter the value of element %d: ", i + 1);
13             ...elements[i] = sc.nextInt();
14         }
15
16         ...int largest = 0;
17         ...for (int i = 0; i < elements.length; i++) {
18             ...if (elements[i] > largest) {
19                 ...largest = elements[i];
20             }
21         }
22         ...System.out.printf("The largest number is %d\n", largest);
23
24         ...sc.close();
25     }
26 }
...
> javac AssignmentTwo.java && java AssignmentTwo
Enter the number of array elements: 4
Enter the value of element 1: 27
Enter the value of element 2: 8
Enter the value of element 3: 33
Enter the value of element 4: 11
The largest number is 33
> |
```

Figure 9: Assignment 2 code and output

3. Create a program that accepts the number of array elements as input, also input the elements of array. Then display which numbers are even and which are odd numbers. Examples of program results are as follows:

```
Enter the number of array elements: 6
Enter the value of element 1: 7
Enter the value of element 2: 3
Enter the value of element 3: 5
Enter the value of element 4: 8
Enter the value of element 5: 2
Enter the value of element 6: 1
Even number: 8
Even number: 2
Odd number: 7
Odd number: 3
Odd number: 5
Odd number: 1
```



The screenshot displays a Java program in an IDE. The code on the left defines a class `AssignmentThree` with a `main` method. It uses a `Scanner` to read the number of elements and their values. It then iterates through the array, printing even and odd numbers. The output on the right matches the example provided in the text.

```
basic-programming-practicum > 2022-11-10 > jobsheet-8-array-1 > codes > AssignmentThree.java > Assign
1  import java.util.Scanner;
2
3  public class AssignmentThree {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          System.out.print("Enter the number of array elements: ");
8          int elementCount = sc.nextInt();
9
10         int[] numbers = new int[elementCount];
11
12         for (int i = 0; i < elementCount; i++) {
13             System.out.printf("Enter the value of element %d: ", i + 1);
14             int input = sc.nextInt();
15             numbers[i] = input;
16         }
17
18         for (int i = 0; i < numbers.length; i++) {
19             if (numbers[i] % 2 == 0) {
20                 System.out.printf("Even number: %d\n", numbers[i]);
21             }
22         }
23
24         for (int i = 0; i < numbers.length; i++) {
25             if (numbers[i] % 2 != 0) {
26                 System.out.printf("Odd number: %d\n", numbers[i]);
27             }
28         }
29
30         sc.close();
31     }
32 }
```

```
> javac AssignmentThree.java && java AssignmentThree
Enter the number of array elements: 6
Enter the value of element 1: 7
Enter the value of element 2: 3
Enter the value of element 3: 5
Enter the value of element 4: 8
Enter the value of element 5: 2
Enter the value of element 6: 1
Even number: 8
Even number: 2
Odd number: 7
Odd number: 3
Odd number: 5
Odd number: 1
> |
```

Figure 10: Assignment 3 code and output