

Assignment 5 - Ten Array Methods

CSE 110 Principles of Programming with Java
Spring 2021

Due April 10th 2021, 11:59PM Arizona Time

1 Assignment Objectives & Requirements

1.1 Assignment Objectives

After completing this assignment the student should be able to:

- Declare and instantiate arrays
- Access array elements by index
- Use loops and decisions to manipulate arrays and array elements
- Write methods that manipulate arrays
- Write methods that take array arguments
- Write methods that return array references

1.2 Assignment Requirements

For this assignment you are given the following file -

Assignment5.java (you must complete this file).

2 Problem Description and Given Information

Within the **Assignment5.java** file, you must define the following static methods. In the **main()** method, you may write any code that wish to test the methods you have been asked to define.

1. Write a public static method named **printArray()**, that takes two arguments. The first argument is an Array of **int** and the second argument is a **String**. The method should print out a list of the values in the array, each separated by the value of the second argument.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

printArray(myArray, ",") will print out **1,22,333,400,5005,0**

printArray(myArray, "-") will print out **1-22-333-400-5005-0**

2. Write a public static method named **getFirst()**, that takes an Array of **int** as an argument and returns the value of the first element of the array.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

getFirst(myArray) will return **1**

3. Write a public static method named **getLast()**, that takes an Array of **int** as an argument and returns the value of the last element of the array.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

getLast(myArray) will return **0**

4. Write a public static method named **getAllButFirst()**, that takes an Array of **int** as an argument and creates and returns a new array with all of the values in the argument array except the first value.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

getAllButFirst(myArray) will return an array of **int** with these values **{22,333,400,5005,0}**

5. Write a public static method named `getIndexOfMin()`, that takes an Array of `int` as an argument and returns the index of the least value in the array.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

`getIndexOfMin(myArray)` will return `5`

6. Write a public static method named `getIndexOfMax()`, that takes an Array of `int` as an argument and returns the index of the maximum value in the array.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

`getIndexOfMax(myArray)` will return `4`

7. Write a public static method named `swapByIndex()`, that takes three arguments. The first argument is an Array of `int`, and the second and third arguments are `int` indexes. This method will swap the values at the two given index arguments in the array and display the array after the swap is complete.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

`swapByIndex(myArray,0,5)` will display `{0, 22, 333, 400, 5005, 1}`;

8. Write a public static method named `removeAtIndex()`, that takes two arguments. The first argument is an Array of `int`, and the second is an `int` index. This method creates and returns a new array with all of the values in the argument array except the value at the argument index.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

`removeAtIndex(myArray,1)` will return an array containing `{0, 333, 400, 5005, 1}`;

9. Write a public static method named `insertAtIndex()`, that takes three arguments. The first argument is an Array of `int`, the second is an `int` index, and the third is an `int` value. This method creates and returns a new array with all of the values in the argument array including the third argument value inserted at the second argument index.

Example:

```
int myArray[ ] = {1, 22, 333, 400, 5005, 0};
```

`insertAtIndex(myArray,1,12)` will return an array containing `{0, 12, 22, 333, 400, 5005, 1}`;

10. Write a public static method named `isSorted()`, that takes an Array of `int` as an argument. This method should return the `boolean` value `true` if all the element values in the array are in ascending order; otherwise the method should return the `boolean` value `false`.

Example:

```
int myArray[ ] = { 22, 323433, 400, 524005, 0};
```

`isSorted(myArray)` will print out `false`

3 Method Template

Here is a template that you may use or refer to when defining your methods. All of your five methods will follow this template; you must provide the components designated by the angle brackets `< >`.

```
static < returnType >< methodName > (< parameters >)
{
    < methodBody >
}
```

4 To Do

After reading the problem description, you may follow the given steps to starting your assignment -

1. Create a new project in your IDE called **Assignment5**
2. Create a new source file called **Assignment5.java** inside the project
3. Copy the contents of the source file provided for this assignment into the one created by you
4. Follow the comment sections along with the requirements listed in section 2 to complete your code
5. Compile and run your program to check for errors
6. Make sure you submit your source code file **Assignment5.java** to the submission link by the deadline

5 Submission Guidelines

Please follow the guidelines listed below prior to submitting your source code file **Assignment5.java** on Canvas -

1. Make sure that your source code file is named **Assignment5.java** prior to submitting.
2. Make sure that your input and output matches the format shown in section 2
3. Make sure that you have completed the comment section at the top of the source code file
4. Submit your **Assignment5.java** file only to the Canvas link for Assignment 5 by **April 10th 2021, 11:59PM Arizona Time**.

6 Grading Rubric

Criteria	Points
<i>All required files are submitted</i>	10
Each file includes the comment head section completed.	
<i>Code is neat and well organized</i>	10
Good naming conventions for all identifiers Good use of whitespace Descriptive comments Partial credit can be awarded	
<i>Code compiles with no syntax errors</i>	20
No Partial credit can be awarded No credit will be awarded for structure or logic if your code does not compile	
<i>Code passes structure tests</i>	30
Code outputs results	
<i>Code passes logic tests</i>	30
Partial credit is awarded based on number of tests passed No credit will be awarded for logic if your code does not pass all structure tests	
TOTAL	100
