

Exercise 9

Task 1 – Creating a Class with a One-Dimensional Array of Primitive Types

The number of vacation days that an employee at DirectClothing, Inc. receives is based on the number of years that the employee has worked for DirectClothing, Inc. Following is the vacation scale for DirectClothing, Inc.

Number of years of employment	Number of days of vacation
Up to 1 year	10
1, 2, or 3 years	15
4 or 5	20
6 or more years	25

In this task, you write a class called `VacationScale` containing an array of seven elements. Assume that the number of years of employment is relative to the position of the element in the array. Each element contains the number of days of vacation corresponding to the number of years of employment.

1. Create a new Java project named `Exercise9`.
2. Write a class called `VacationScale` that contains a member variable called `vacationDays` of type `int[ ]`. Each element will contain the number of days of vacation relative to its index in the array. For example, index 0 represents 0 years of service, so `vacationDays[0]` will contain the number 10, and index 4 represents 4 years of service, so `vacationDays[4]` will contain the number 20.
3. Create a public method called `setVacationScale` with no argument and no return type. Initialize `vacationDays` to size seven and set values in all the elements of the `vacationDays` array.
4. Create a public method called `displayVacationDays` that returns nothing, accepts an (`int yearsOfService`) argument, and displays the number of days of vacation the employee receives. For example, if a 1 is passed to `displayVacationDays`, then the number 15 is displayed.

Hint: You can use a variable within the square brackets when retrieving values from an array. For example

```
vacationDays [ yearsOfService ];
```

5. Add a test class called VacationScaleTest and add codes below.

```
public class VacationScaleTest {  
    public static void main(String args[]) {  
        VacationScale myVacationScale = new VacationScale();  
        myVacationScale.setVacationScale();  
        myVacationScale.displayVacationDays(1);  
        myVacationScale.displayVacationDays(5);  
        myVacationScale.displayVacationDays(10);  
    }  
}
```

6. Run the test class, the output of your program should be similar to

```
Your vacation is: 15  
Your vacation is: 20  
Your vacation is: 25
```

Task 2 – Creating and Using Two-Dimensional Arrays

The objective of this exercise is to create and access a two-dimensional array.

This exercise is based on the scenario of a classroom. A classroom has 12 desks arranged in a rectangular grid comprising of 3 rows and 4 columns. Students are allocated a desk at the position found vacant first, by traversing each row.

Figure below shows the class map as a grid. Each cell represents a desk. Each cell contains the co-ordinates of the desk position in the class map.

XXXXX	Col 1	Col 2	Col 3	Col 4
Row 1	0,0	0,1	0,2	0,3
Row 2	1,0	1,1	1,2	1,3
Row 3	2,0	2,1	2,2	2,3

In this exercise, you create a Java class to represent a classroom map, allocate desk positions to the students of a class, display the desk map of the class, and search for the allocated desk position of a student.

1. Create a new Java class ClassMap.
2. In the class, declare two member variables, roomNo of type int and deskArray of type String[][].
3. Create an argument constructor that accepts an id of type int in the constructor.
 - a. Assign the value of the argument to the member variable roomNo.
 - b. Initialize the size of the deskArray to [3][4].
4. Create a method by the name setDesk that accepts a String argument and return no value. This argument will contain the name of the student.
5. In the setDesk method, write code to perform the following tasks.
 - a. Traverse the deskArray array to identify the first vacant element in it. You can use a nested for loop for this purpose. For example:

```
for ( int row = 0; row < 3; row++ ) {  
    for ( int col = 0; col < 4; col++ ) {  
        if ( deskArray[row][col] == null ) {
```

- b. Assign the student's name to the vacant element.
 - c. Print the position of the desk for the student and exit out of the loops.
6. Create a public method by the name displayDeskMap with no argument and no return type. In the method, write code to traverse through the deskArray and print the names in each element of the array, such that the names are displayed in grid form.
 7. Create a method by the name searchDesk that accepts a String argument and returns no value. The argument represents the name of the student whose desk position has to be searched from the deskArray.
 8. In the searchDesk method, write code to do the following.
 - a. Create a nested for loop to traverse through the deskArray array.
 - b. If the array element is not null, compare the value of the method argument with the element. For example:

```
if ( deskArray[row][col] != null && deskArray[row][col].equals(name) == true ) {
```

- c. Print the position of the desk if the names are equal.
 - d. Print an error message if the name is not found in the deskArray array.
 - e. Use the break statement to branch or exit out of the loops wherever required.
9. Create a new Java class called ClassMapTest and add codes below.

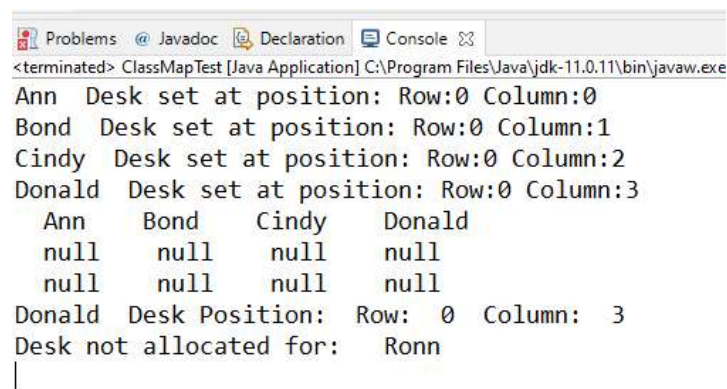
```
public class ClassMapTest{
    public static void main(String args[]){

        // creating ClassMap instance
        ClassMap cm =new ClassMap(1);

        // Allocating position to one student
        cm.setDesk("Ann");
        //Allocating position to another student
        cm.setDesk("Bond");
        //Allocating position to third student
        cm.setDesk("Cindy");
        //Allocating position to fourth student
        cm.setDesk("Donald");
        //Displaying the map of the class
        cm.displayDeskMap();
        //Searching the desk of a student
        cm.searchDesk("Donald");
        //Searching the desk of another student
        cm.searchDesk("Ronn");

    } //end main
} //end class
```

10. Study the main method and check if all the methods of ClassMap class has been invoked. You can add your own test too.
11. Run the test program and verify the output.



```
<terminated> ClassMapTest [Java Application] C:\Program Files\Java\jdk-11.0.11\bin\javaw.exe
Ann Desk set at position: Row:0 Column:0
Bond Desk set at position: Row:0 Column:1
Cindy Desk set at position: Row:0 Column:2
Donald Desk set at position: Row:0 Column:3
Ann Bond Cindy Donald
null null null null
null null null null
Donald Desk Position: Row: 0 Column: 3
Desk not allocated for: Ronn
|
```

THE END.