

Tutorial 10: Arrays and Reference Parameters

This tutorial will provide practice at using arrays to store data and manipulating that data, and provide practice at passing arrays to methods.

Task 1: Reverse



Write a program to read in 10 integers and store them in an array. After the integers have been input, the program should print them out in reverse order.

Hints: the contents of the array are not modified use a `for` loop
and decide where to start and where to finish

Step 1: Create a NetBeans project

- create a new project called `ReverseProj` in a folder called `T10`

Step 2: Write source code

- add a new file called `Reverse` to the `ReverseProj` project
- write code to read in and store the integers, then print them in reverse order

Step 3: Test your program and take screen shots

- run your program
- take a screen shot of the output and save it in your project folder as `Reverse.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `Reverse.jpg` from step 3, containing a screen shot of the output

Task 2: SmallestInArray



Write a program to read in 10 integers and store them in an array. After the integers have been input, the program should find and output the smallest value and its position in the array.

NB: you must store the values in an array, rather than finding the smallest value as the numbers are input

Step 1: Analyse the problem

- produce pseudo-code for the `SmallestInArray` program and store it in a file called `SmallestInArrayPseudo.doc`

Step 2: Create a NetBeans project

- create a new project called `SmallestInArrayProj` in a folder called `T10`

Step 3: Write source code

- add a new file called `SmallestInArray` to the `SmallestInArrayProj` project
- using the pseudo-code from step 1, write code to output the value and position of the smallest element

Step 4: Test your program and take screen shots

- run your program with the following sets of data:
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
- take screen shots of the output from both runs and save them in your project folder as `SmallestInArray1.jpg` and `SmallestInArray2.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `SmallestInArrayPseudo.doc` from step 1 containing the pseudo-code for the find smallest program
- `SmallestInArray1.jpg` and `SmallestInArray2.jpg` from step 4, containing screen shots of the output

Task 3: AboveAverage



Write a program to analyse student results and output the names and marks of those whose marks are above average.

The program should prompt the user to enter the number of students to process followed by the name and mark for each student. The names should be stored in an array of strings and the marks in an array of integers. The program should calculate and output the average mark, followed by the name and mark for all students whose marks are above that average.

Step 1: Create a NetBeans project

- create a new project called `AboveAverageProj` in a folder called `T10`

Step 2: Write source code

- add a new file called `AboveAverage` to the `AboveAverageProj` project
- write code to output the details of all students with above average marks

Step 3: Test your program and take screen shots

- run your program
- take a screen shot of the output and save it in your project folder as `AboveAverage.jpg`
- run your program again with 0 students and ensure that it does not crash
- take another screen shot of the output with 0 students and save it in your project folder as `NoStudentsAboveAverage.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `AboveAverage.jpg` and `NoStudentsAboveAverage.jpg` from step 3, containing screen shots of the output

Task 4: AllEven



Write a program to prompt the user to enter the size of an array followed by that number of integers which are to be stored in the array. The program should then create a new array containing only the even numbers from the original input, and display its contents. The program should be implemented using 3 methods: `read`, `getEven`, `print`

Hints: you should count the number of even numbers before you create the new array to hold the even numbers

Step 1: Analyse the problem

- using the examples shown in the lecture, produce tables showing the parameters, return types and descriptions for each of the methods and store them in a Word document called `allEvenMethods.doc`
- produce pseudo-code for the `getEven` method and store it in a Word document called `AllEvenPseudo.doc`

Step 2: Create a NetBeans project

- create a new project called `AllEvenProj` in a folder called `T10`

Step 3: Write source code

- add a new file called `AllEven` to the `AllEvenProj` project
- write code to create and print the contents of the new array containing the even numbers that were input by the user

Step 4: Test your program and take screen shots

- run your program with all even numbers
- run your program with all odd numbers
- take screen shots of the output from both runs and save them in your project folder as `AllEven.jpg` and `AllOdd.jpg`

Portfolio requirements

- The NetBeans project for this completed task

- `AllEvenMethods.doc` containing details of the method interfaces from step 1
- `AllEvenPseudo.doc` containing pseudo-code from step 1
- `AllEven.jpg` and `AllOdd.jpg` from step 4, containing screen shots of the output

Task 5: Band



A program is required to store details for a band comprising a drummer, a bass player, lead guitarist, keyboard player and a vocalist.

Create a `Member` class to represent the band members, storing their stage name and instrument. Create an array of 5 band members and store their details.

Step 1: Create a NetBeans project

- create a new project called `BandProj` in a folder called `T10`

Step 2: Write the `Member` class

- add a new file called `Member` to the `BandProj` project
- using the above information implement the private variables and appropriate public get and set methods for the `Member` class

Step 3: Create the `Band` application that uses `Member` objects

- add a new file called `Band` to the `BandProj` project
- create an array of 5 `Member` objects and read their details from the keyboard
- write a method to display the band members

Step 4: Run your program and take screen shots

- run your program
- take screen shots of the output and save it in your project folder as `Band.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `Band.jpg` from step 4, containing screen shots of the output

Task 6: Battleships



You may work on this problem with another student in your tutorial group.

Write a program for (partially) playing the game of Battleships.

The program maintains its own (hidden) board as a 12x12 array of integers. If a ship occupies a grid location, 1 is stored in the array, otherwise zero is entered. You should hard-code the values in the array for this task.

The program repeatedly prompts the user to enter the coordinates of a shot, and the program reports:

- “Hit” if the nominated element in the program’s array contains a 1;
- “Miss” otherwise.

User shots are displayed on a board based on a 12x12 array of characters. In this board, the program records “H” for hits, and “X” for misses, and after each shot the board is printed so the user can see the accumulation of shots.

Here is an example:

```

Output
Debugger Console  BattleshipsProj (run)
Y
Enter the x coordinate of your shot (1-12)
5
Enter the y coordinate of your shot (1-12)
2

  1 | | | | H | | | | | | | |
  2 | | | | X | | | | | | | |
  3 | | | | | | | | | | | | |
  4 | | | | | | | | | | | | |
  5 | | | | | | | | | | | | |
  6 | | | | | | | | | | | | |
  7 | | | | | | | | | | | | |
  8 | | | | | | | | | | | | |
  9 | | | | | | | | | | | | |
 10 | | | | | | | | | | | | |
 11 | | | | | | | | | | | | |
 12 | | | | | | | | | | | | |
    1  2  3  4  5  6  7  8  9 10 11 12
Do you want another shot? (y/n) :>

```

You are to demonstrate your ability to use methods, value parameters, reference parameters, returning values, and returning references.

[Portfolio requirements](#)

- The NetBeans project for this completed task
- Image files containing screen shots of your application's output