Arrays

Introduction

This unit covers how to use data structures that store more than one value.

One Dimensional Arrays

Arrays are used to store multiple values of a single type into one variable.

Information about Arrays

- Arrays can store primitives or objects.
- When an array is created its size is set and cannot be changed later.
- Arrays store and access values based on **indexes**, locations where values are stored.
- Each value stored by an array is called an **element**.
- The first index in an array is at index 0
- The last index is (number of elements stored) -1.
    - For example, an array that stores 8 elements would have a starting index of 0 and ending index of 7.
- When an array is created its values will be initialized to 0 automatically, false for boolean arrays.

Creating a One Dimensional Array

Creating a 1D array when the size is known but the values are not.

type[] name = new type[size];

Example:

```
int[] numbers = new int[5];
// would store 5 integer values
```

Creating a 1D array when the size and values are known

type[] name = {value,value,value,value,…, value};

Example:

```
String[] names = {“Joe”, “Bill”, ”Jane”};
/* would store “Joe” at index 0, “Bill” at index 1
and “Jane” at index 2*/
```

Getting the size of a 1D array

name.length

Example:

```
System.out.println(names.length);
```

Output:

```
3
```

Assigning or changing values in an array after it has been created

name[index] = value;

Example:

names[1] = “Tina”;

Accessing the value at an index

name[index]

Example:

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

Output:

Joe
Tina
Jane

Printing all the values

```
for(int x = 0; x < name.length; x++)
    System.out.println(name[x]);
```

Example:

```
for(int x = 0; names.length; x++)
        System.out.println(names[x]);
```

Output:

Joe
Tina
Jane

Two Dimension Arrays

Two dimensional arrays are arrays that store arrays. The data in the array is accessed with row and column values.

Creating a 2D array when the size is known but the values are not

type[][] name = new type[number of rows][number of columns];

Example:

```
char[][] ticTacToeBoard = new char[3][3];
```

Creating a 2D array when the size and values are known

type[][] name = {{value,value,value,value,…,value,value},
{value,value,value,value,…,value,value},
…
{value,value,value,value,…,value,value},
{value,value,value,value,…,value,value}};

Example:

```
int[][] numGrid = {   {2,3},
                      {7,4},
                      {1,3},
                      {4,6}};
```

Getting the number of rows that a 2D array has

name.length

Example:

```
System.out.println(numGrid.length);
```

Output:

4

Getting the number of columns that a 2D array has

name[0].length

Example:

```
System.out.println(numGrid[0].length);
```

Output:

2

Assigning or changing values in an array after it has been created.

name[row][column] = value;

Accessing the value at an index

name[row][column]

Example:

```
System.out.println(numGrid[3][1]);
numGrid[3][2]=1;
System.out.println(numGrid[3][1]);
```

Output:

6
1

Printing all the values

```
for(int r = 0; r < name.length; r++)
{
    typeOfArray[] row = name[r];
    for(int c = 0; c < row.length; c++)
    {
        System.out.print(name[c] + " ");
    }
    System.out.print("\n");
}
```

Example:

```
for(int r=0; r<numGrid.length;r++)
{
    int[] row = numGrid[r];
    for(int c = 0; c <row.length; c++)
        System.out.print(row[c]+ " ");
    System.out.print("\n");
}
```

Output:

```
2 3
7 4
1 3
4 1
```

For Each Loop

A for each loop is a special for loop for going through all the items in a data set.

Format for a for each:

```
for(typeOfDataInArray variableName: arrayName)
    // code using the variableName
```

The loop will run once for every item in the array and each time variableName will store a different value from the array.

Example:

```
int nums[] = {4,8,7};

int sum = 0;
for(int x: nums)
    sum+=x;

System.out.println("The sum of the data set is "+ sum);
```

Output:

The sum of the data set is 19

Example:

```
int[][] numGrid = {     {2,3},
                        {7,4},
                        {1,3},
                        {4,6}};
for(int[] row: numGrid)
{
        for(int num: row)
        {
                System.out.println(num + " ");
        }
        System.out.println();
}
```

Output:

2 3
7 4
1 3
4 6

Turning a String into a char Array

The following code will create a char array containing the same text as a String.

Format for getting a char array from a String:

char[] arrayName = StringName.toCharArray();

Example:

```
String s = "This is a string";
char[] textArray = s.toCharArray();
```

Arrays Class

The Arrays class has methods to perform operations on arrays.

Math Methods

Method	Description
sort(arrayName)	Sorts the given array.
toString(arrayName)	Prints the given 1D array
deepToString(arrayName)	Prints the given multi-dimensional array

Blue Pelican Sections

Lesson 18
Lesson 35

Terms

Array	A structure for storing multiple values in a single variable.
Element	A single value stored by an array.
Index	A location in an array, where a value is stored.