Lesson 6: Basic Arrays

1. An *array* is a collection of multiple values in a single variable with similar data types by using an index.

2. Index- a number that represents a position in a collection/an array; starts at 0
3. Elements- individual values in an array

4. Arrays can contain any type of value (base types or objects), but you can't store different types in a single array. You can have an array of integers, or an array of strings, or an array of arrays, but you can't have an array that contains, for example, both strings and integers.

   Declare w/ values:
   String names[] = {"Chris","Sam","Peter"};

   Declare w/o values:
   String [] array = new String [100];

Reading Array Element
identifier[index];
names[0];

Assigning Array Element
identifier[index] = value;
names[0] = "Jaymar"

```
public static void main(String[] args) {
    String studentNames[] = {"David","Alenere","Jaymar","Ace","Jasfer"};
    System.out.println(studentNames[3]);
```

```
String studentNames[] = {"David","Alenere","Jaymar","Ace","Jasfer"};
studentNames[0] = "Emman";
System.out.println(studentNames[0]);
```

5. To *create an array* in Java, we follow the following syntax:
   dataType - it can be primitive data types like int, char, double, byte, etc. or Java objects
   datatype [] arrayName;

6. *arrayName* - it is an identifier

7. In Java, we can declare and allocate the memory of an array in one single statement. For example,
   double [] data = new double[10];

8. In Java, we can *initialize arrays* during declaration. For example,
   int [] age = {12, 4, 5, 2, 5};

9. We can access the element of an array using the index number. Here is the syntax for accessing elements of an array:
   Array[index]

10. In Java, we can also loop through each element of the array. For example,

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

// create an array
int [] age = {12, 4, 5};

// loop through the array using for loop
System.out.println("Using for Loop:");
for(int i = 0; i < age.length; i++) {
System.out.println(age[i]);
```

11. A multidimensional array is an array of arrays. Each element of a multidimensional array is an array itself.

12. Iterating arrays- reads every element inside an array and does something with it.

13. Accessing arrays

```
Scanner s = new Scanner (System.in);
Names[0] = s.next line();
```

14. BREAK Keyword- used to break out of a loop statement