**Unit 6 Cheat Sheet**

* Arrays represent collections of related data all of the same data type.
* The size of an array is established at the time of creation and cannot be changed.
* Arrays can store either primitive data or object reference data.
* When an array is created using the keyword new, all of its elements are initialized with a specific value based on the type of elements:
* Elements of type int are initialized to 0
* Elements of type double are initialized to 0.0
* Elements of type boolean are initialized to false
* Elements of a reference type are initialized to the reference value null. No objects are automatically created.
* Initializer lists can be used to create and initialize arrays.
* Square brackets ([ ]) are used to access and modify an element in an array using an index. The indexed array variable, for example array[index], can be used anywhere a regular variable can be used, for example to get or assign values.
* The valid index values for an array are 0 through one less than the number of elements in the array, inclusive. Using an index value outside of this range will result in an ArrayIndexOutOfBoundsException being thrown.

Declaring arrays:

Text

Description automatically generated with medium confidence

\*Can be done similarly for any data type or class.

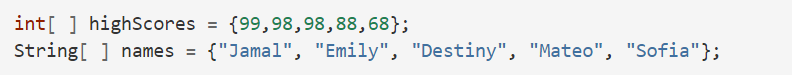
Declaring & Creating arrays:

(with new)

Text

Description automatically generated

(with initializer list)



Accessing and modifying arrays:

To access the elements in an array, we use an indexed array variable which is the array name and the index inside of square bracket [ ].

arrayName[index]

For example:

System.out.println(array[2]);

Or

int a=array[0];

Traversing arrays:

* You can use loops to traverse through arrays (go through all the elements).
* To traverse from the last index to first index use arrayName.length – 1 as your initial index
* Make sure to put in conditions in your for loop to avoid errors (traversing through indexes that are out of bounds)
* Remember arrays start at the 0th index

For each:

\*To set up a for-each loop, use for (type variable : arrayname) where the type is the type for elements in the array, and read it as “for each variable value in arrayname”.

Example:

Graphical user interface, text

Description automatically generated

\*important: Only the variable in the loop changes, not the real array values.