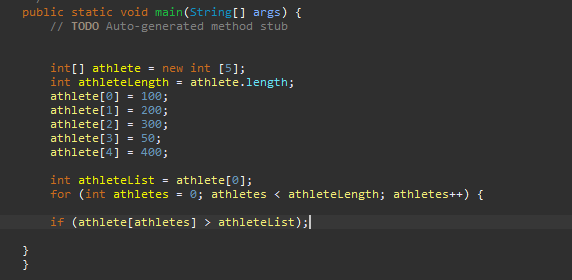
## Module 4: Arrays, Methods, and Memory

#### Arrays:

* An array is a way to store more than one value – it’s a data structure in a form like a table.

### Array Rules:

* Array size is fixed once designated.
* All data types within array must be the same.
* Use brackets [] when referring to values on an array.
* Arrays have scope just like variables.
* Data within the array is associated with an address.



An array, it contains 5 spaces each able to hold a value, all of them are integers.

Declaring & Initializing Arrays:

* Arrays are formed in a reference variable.
  + A reference variable is an address to a certain part of the array.
  + This reference variable as well as an array, is an object.
* Declaration

*Datatype[] arrayName = new dataType [arraySize]*



* Datatype [] and new keyword
  + New – The keyword used in declaring an *object*.
  + Datatype [] arrayName – This will indicate the declaration of an array under a certain name – this is called a reference variable.
  + Datatype [arraySize] – The data type within the array size must be an integer, memory will be allocated for the array size.
* Initialization

The initialization of an array resides within the array size. The integer value of the array size is allocated within the memory

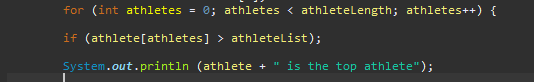
* Memory can also be allocated dictated by a number list.



This indicates an array in the form of a number list. These 5 values, (100,200,300,400,500) will be allocated in memory.

Arrays In Loops:

* When using arrays in loops, its best to use for loops.
  + For loops are loops that can be ran a set number of times.



A for loop involving an array of athletes and finding the highest value.

* This loop will run if it has run less than the length of the array.
* This will run a check for every spot within the array comparing the values.
* Once it finds the highest value, it will print which spot in the array it is, and its value

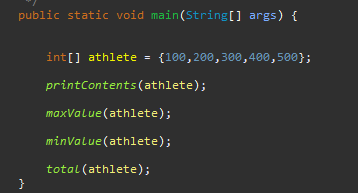
Stack and Heap:

### Memory Storage in Stack:

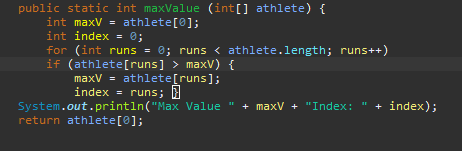
* Methods are stored within the stack
* Local variables are stored within the stack
* Addresses or reference variables are stored within the stack
* Primitive datatypes are stored on the stack

### Heap:

* Objects are stored on the heap.
* Arrays are stored within the heap.
* Classes are stored within the heap.
* Non-primitive data types are stored on heap



The array, athlete, is an object, and the data list assigned to it is allocated. – In the main method, it is stored within the heap and utilizes methods.

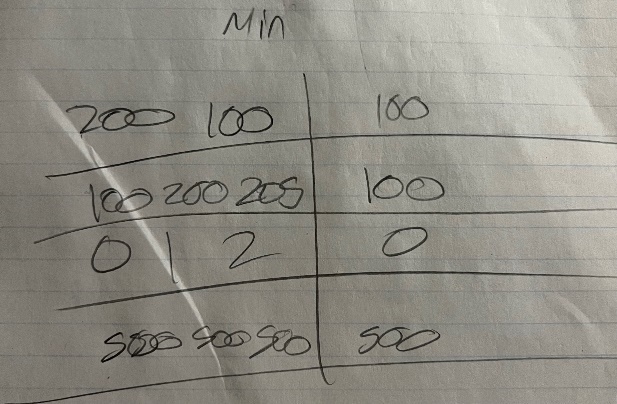


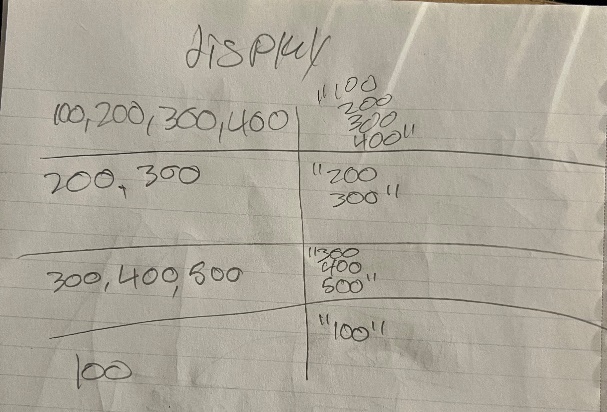
A method used to determine max value, this is stored on the stack, and after code is ran, it will dissipate.

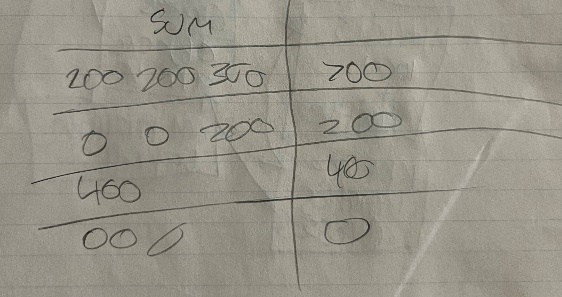


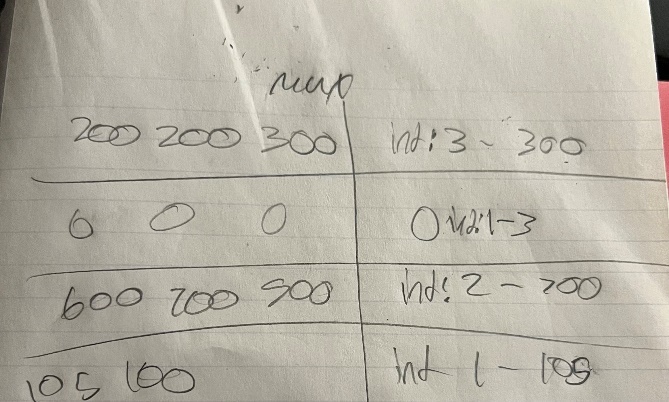
A reference variable. This variable references an address to a certain point on the array, this is stored on the stack.

## Pseudo Code Examples: Top Athlete

1. A piece of paper with writing on it

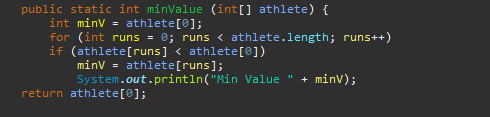
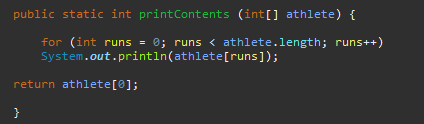
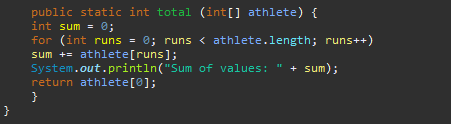
   AI-generated content may be incorrect.
2. A piece of paper with writing on it

   AI-generated content may be incorrect.
3. A writing on a piece of paper

   AI-generated content may be incorrect.
4. A piece of paper with writing

   AI-generated content may be incorrect.

## Pseudo Code Translation: Top Athlete

1. 
2. 
3. 
4. A screen shot of a computer code

   AI-generated content may be incorrect.

#### 2-D Arrays:

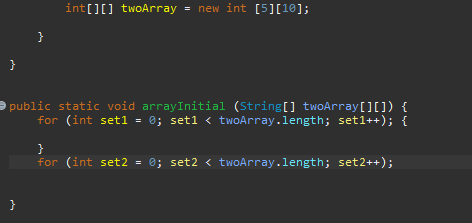
* Essentially an array inside of an array.
* Like a spreadsheet – rows and columns

Declaration:

* Int [] [] array = new int [size] [size]
  + [] [] – indicates 2-D array
  + [size 1] – rows [size 2] = columns

Initialization:

* Initialize with for loops
  + Works the same as a single array, but with a nested second for loop



Two nested for loops: deal with the first array [5] and second array [10]

#### Write and Read in Files:

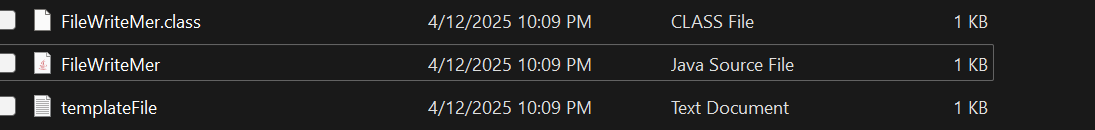
First, to read or write in files, the java.io package needs to be imported.

*Reading and Writing:*

* *A scanner will be needed to read files, like a keyboard input.*
  + *Remember to close scanner*
* *A file must be in the same folder as the java file to be referenced.*
* *Files can be created; they are the datatype of object and use the “new” keyword.*
* *File names are strings*



Writing in a file

**

Files in same folder



Importing io and scanner package, both needed for the above file write

#### Passing Arrays, Primitive Data, Objects, and Methods:

Passing Primitive Data Types:

* Pass by value
  + Essentially, creating a copy of the value and passing it to a method.
  + Int numbers = 0 – datatype name = copy of value

Passing Objects:

* Pass by reference
  + Similar to pass by value, but instead, it uses an address as a value like an index
  + numbersArray[5] – array[index 5]

Stack Memory – Methods:

* Methods are stored on the stack
* They are called upon by parentheses – method(value) – updateRandomValue(number)\
* They are stored on the stack, and are allocated in memory as the code runs
* Once the code stops, the method then dissappears

