

Most all "types" and functions in java are class types and functions.
In high level concept, they are basically the same thing as C++'s Standard Template Library stuff
(e.g. `std::vector`, `std::string`, `std::cout`, ...)

However you started with all that stuff given to you (mostly by automatic defaults of Java)

C has none of it.

C++ by itself has very little of it.

(both have libraries... but we will put that aside for the moment)

(they are not "automatically" available)

From basics you have:

```
int i;
char ch;
bool boo;    <-- C technically does not have bool, C++ does
```

From there you can make arrays

```
int A[50];
char C[50];
bool B[50];
```

From that you have pointers

```
int *eye_ptr;
char *letter_pointer;
bool *truth_pointer;
```

They can point to single values

(have the the memory address of a single other variable)

```
eye_ptr = &i;
letter_ptr = &ch;
truth_pointer = &boo;
```

Or they can point to an array (a block of contiguously allocated variables)

(by having the memory the address of the first thing in an array)

```
eye_ptr = A; or eye_ptr = &(A[0]);
letter_ptr = C; or letter_ptr = &(C[0]);
truth_ptr = B; or truth_ptr = &(B[0]);
```

And that's pretty much it (some details and similar types omitted/skipped)

That's all you get for C and C++ as far as built-in types.

From there you have to make your own types.

C allows this to be done using:

typedef, enum, and structs (and some I will skip for brevity)

C++ adds the option of using:

class (and some I will skip for brevity)

Unit1:

- Lots of C++ stuff about syntax and coding up things
- Focus was how to make a C++ class
- A couple sorting and searching things along the way
- Some stuff on dynamic memory allocation and deallocation
- A little on static and dynamic arrays
- A little on linked lists
- An intro to Big-Oh
- Likely some other points

Here you saw

- how to use basic types
- how to create classes (new types)
- how to use arrays (mostly static, but mixed with dynamic memory allocation)
- how to make linked lists (briefly)

Unit2:

- C++ usage continues (a quick file input/output diversion)
- Lots of various on Sorting and Searching
- Lots on Stacks, Lists, Queues, Dequeues
- Emphasis:
 - Abstract Data Type Descriptions
 - Various ways to implement using other data types

Here we started back with

- definition of a linked singly linked list
- how to create linked lists with basic struct or class

- definition of an ADT stack
- how to implement a ADT stack with an singly linked list

- definition of a doubly linked list
- definition of an ADT queue
- how to implement a queue with an doubly linked list

- a dynamic array ADT
- a vector ADT

- a diversion on ADT iterators

- an ADT List
- how to implement an ADT List with a singly linked list
- how to implement an ADT List with a doubly linked list

- a diversion on ADT position

an ADT sequence

how to implement an ADT sequence with an ADT array

how to implement an ADT sequence with an ADT list

And moving on to Algorithms

BubbleSort

MergeSort

(both sort of assuming ADT sequence type exists and can be used)

And a slight reminder of the Big-Oh stuff

And that covers most of the course so far