C	Vs	C++
* printf & scanf		* cin & cout
* Procedure Oriented		* Object Oriented
* Uses structures & unions		* Uses classes & objects
* No built-in bool or string		* Built-in bool & string
* No Namespaces		* Uses Namespaces
* malloc, calloc, realloc, free for Dynamic Memory Allocation		* new & delete for Dynamic Memory Allocation
* No built-in Data Structures		* Has the built-in STL Library
* (while, do-while, for)		* All loops plus (for each loop) (enhanced for loop)

public accessed outside directly

```
Struct Employee {
        char name[20];
        int age;
        char company[20];
};
```

struct Employee e1;

Access Specifiers

public private protected

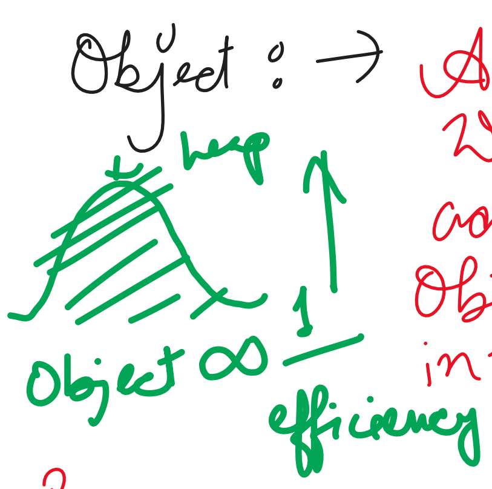
```
Class Employee {
        string Name;
        int Age;
        string Company;
};
```

Objects

We cannot access outside directly (private)

class :→ It is a blueprint or template to create objects / instances / references.
It has no memory of it's own.
It is actually used to represent some real world entities ex: Student, Employee, Car, Rat.

Object :→ An object is an instance of a class. With the help of the object, we can access all the data inside the class. Objects have memory. They are stored in the heap memory of the OS.



Blueprint

```
class Employee {
        string Name;
        int Age;        } → properties / attributes / fields
        string Company;
};
```

Employee e1; object / instance

* Methods (Functions): In C++ or Java, the functions are known as methods because they actually describe / depict the behaviour of each object.

```
class Student {
    speak ();
    read ();     } methods.
    write ();
}
```

difference b/w class & struct : (DRY)

* All the data members inside a class are private by default. They cannot be accessed outside directly.
* All the data members inside a struct / union are public by default. They can be accessed outside directly.

Constructor :→ ① It is a special method inside the class used to (initialize) objects (instantiate)

② It is written by writing the name of the class followed by parentheses ( ).

```
class Student {

    Student ( ){

    }                constructor

};
```

③ There are two types of constructors :→
ⓐ default constructor : If we don't create a constructor, the compiler will automatically generate one for us. This is called the default Constructor.
ⓑ Parameterised Constructor : If we create our own constructor, the default constructor is deleted. This one is called the parameterised constructor.

* There can be multiple constructors inside a class. It is called constructor overloading. (Polymorphism)

* A constructor is always public. Otherwise we can't create objects.

Destructor :→

* When we create objects, constructor is called & the heap memory is increased immediately.
* Therefore, we must reduce the heap size to improve system efficiency.
* We we the destructor to destroy objects.
* Syntax : ~ ClassName ( );
* It is automatically called at the end of program execution to destroy all the objects.
* Note : There can be only 1 destructor in a class.

Basic OOPs ✓, Constructor ✓, Destructor ✓, Methods ✓,

Copy Constructor, this pointer, name clash
4 Pillars
Exception Handling

STL library
DMA C++