C++ STLs

"Like a BOSS"

video - 5 ⟸

Comparators

- It is a function or an object that is used to compare elements.

- Used to decide the order of elements inside a Data Structure. (vector, set, map, heap)

They can be implemented in different ways :-

① Function Pointers - A function that implements the logic of comparison.

Example → sort()

```
vector <int> vec = {3, 2, 1, 6}
```

- Sort (vec.begin(), vec.end()) ; //Default Asc

```
Sort (vec.begin(), vec.end(), myComparator) ;
```

```
bool myComparator( int a, int b) {
        if (a > b) return True;

        return False;
}
```

2 Function Objects (Functors) :-

An object that can be treated as a function.

Example → sort() ⇚

```
vector <int> vec = {3, 2, 1, 6} ⇚

Sort (begin(vec), end(vec), myComparator());

Class myComparator {
        bool operator() (int a, int b) {
                return a<b;   //Asc.
        }
};
```

3 Lambda Expressions :-

Example → sort( )

vector <int> vec = {3, 2, 1, 6}

```
Sort (begin(sort), end (sort), [ ] (int a, int b){
                return a > b ;
});
```

capture clause

```
[ ] (int a, int b) {
        // Body
        return a > b ;  //Desc.
}
```

```
int main ( )     {
```

```
int   main ( )     {
                 vec = { . . .  }
         int   x = 3 ;
         int   y = 4;
               unor_map mp;
      sort(begin(vec), end(vec), [mp ] (int a, int b)

                 mp{

      }

}
```

```
auto myLamba = [ ] (int a, int b) {
          return  a>b;
   };

Sort (begin(vec), end (vec), myLambda);
```