PART 3 SUMMARY – read and memorize !!!!!!

Copy constructor is called automatically when passing an object by value or returning an object by value.

Copy constructor does not work properly for objects with pointers.

Operator= for obj1=obj2; does not work properly for objects with pointers.

Operator== does not even compile for objects.

Inherit from one class to another instead of re-writing the same thing.

Private things are never passed down except to Friend Classes.

Protected things are never available to the client program.

You can inherit publicly or privately (secretly).

Memory Management – Dynamic Memory Management is Expensive

new in C++ → memory manager will give you a cell of the right size in the heap.

delete in C++ → memory manager will return your cell to his "free-cells list."

Garbage cell == the cell you no longer point to but you did not delete.

GC (language dependent) → grabs garbage cells, and returns them to his "free-cells list."

Dangling pointer == your pointer that is pointing to a deleted cell.

Therefore,

- **If you do not need the cell, delete it. (prevent garbage)**
- **If you need the cell, do not delete it. (prevent dangling pointers)**

Issues:

- **GC using a reference counter can even prevent dangling pointers and can grab cells as soon as nobody points to them.**
- **The memory manager has to prevent leaving many tiny free-cells (fragments) on his list as he gives out cells.**

For Programming:

- **Always pay attention to maintainability and re-usability.**
- **You must be able to program on empress if you want to complete in IPC.**