Course Summary

Object Modelling

- To identify classes and attributes use your common sense
- A usual approach is to identify anything with characteristics. Such a thing should be made class and its characteristics should be members variables
- Normally common nouns become classes
- Proper nouns should be instances of those classes
- Normally verbs are member functions

Constructor

- If you want some code to execute as soon as an object is created, then we can write this code in a constructor e.g. for the purpose of variable initialization
- We can make multiple different constructors in a class as long as their parameters are different
- No matter how many constructors are present, each object will use exactly one of them during creation
- If we don't define a constructor in our class, then compiler-provided default constructor with zero parameters is used for object creation. This means that objects can only be created without passing arguments

Copy Constructor

- If we create a copy of an object, then copy constructor is used
- If we don't define a copy constructor, then a compiler-provided copy constructor is used (but it is powerless).
- There can be only one copy constructor in each class
- We don't need a copy constructor unless there is DMA in our class
- If there is DMA and we don't define a copy constructor, then shallow copy happens i.e. copied object will share memory and value
- If there is DMA and we define our own copy constructor and perform new memory allocation, then deep copy happens
- If we don't' want objects to be copied then we can create our own copy constructor and make it private (to stop object copying)

Abstraction

- In object-oriented programming, we want information and code hiding
- Class member variables are usually kept private
- Class member functions are usually kept public
- To read value of a private variable we usually make a public function (getter function)
- To update value of a private variable we usually make a public function (setter function)

Destructor

- There can only be one no destructor in a class and that is without parameters
- If you want some code to execute when object is being destroyed (going out of scope), then we can write this code in a destructor e.g. deallocating memory, closing file
- If we don't define a destructor then, then compiler-provided default destructor is used

Static

- A static class member variable is common for all objects i.e. if one object changes its value then it will be changed for all other objects as well
- We can access a static member variable or function by using an object or class (scope resolution)
- A static function is one that can only use static members of the class
- We normally use static variables to save values such as revenue or cost

Constants ▪ A constant cannot be changed after it is initialized with a value ▪ In C++, a constant must be initialized in the same line that it is declared ▪ If a constant is class is not given an initial value, then we must give it an initial value by using member initialization list in all the constructors ▪ We can create constants with pointers by either making the pointer itself constant, or by making the data constant, or even by making both pointer and data constants ▪ A constant function is one that cannot modify any member variable of the class ▪ A constant parameter is one that cannot be modified inside the function ▪ A constant object is one that can only be used to call constant function of the class		