C++ Programming – Lecture 10

Inheritqnce

- C++ facilitates code reuse at 2 levels : a) Source code level b) Object code level
- Source code level reuse is done using Template functions and Template classes
- Templates let us write generalized functions / classes and the compiler creates specific functions / classes from it
- For creating specialized functions / classes source code has to be available
- Object code level reuse is done using Containership and Inheritance
- Containership should be used when the two classes have a "has a" relationship
- Inheritance should be used when the two classes have a "like a" relationship
- Containership and Inheritance can be implemented even if source code is not available
- Inheritance terminology : base - derived, parent - child
- Protected members are available in the inheritance chain
- Derived class object contains all base class data
- Derived class object may not be able to access all base class data
- Construction of an object always proceeds from base towards derived
- Base class constructor can be called using baseclassname()
- Inheritance facilitates :

  Inheritance of existing feature : To implement this just establish inheritance relationship

  Suppressing an existing feature : Hide base class implementation by defining same function in derived class

  Extending an existing feature : By either providing brand new functionality or making a combination of new and old functionality
- Base class function can be called from derived class function by using the syntax Baseclassname::Baseclassfunction() ;
- There are 4 types of inheritance : 1) Simple 2) Multi-level 3) Hybrid 4) Multiple
- In multiple inheritance a class is derived from 2 or more than 2 base classes
- In multiple inheritance order of classes in the derived class declaration dictates the order in which the constructors of base classes are called
- In multiple inheritance diamond problem can be avoided by using virtual base classes