# C++ Programming – Lecture 11

## Polymorphism

- Polymorphism has two meanings :

  1) One thing existing in several forms
  2) One action results into different activities

- Virtual functions help implement the second type of polymorphism
- Binding means deciding which function to call
- If binding is done at the time of compilation it is called Early Binding
- If binding is done at the time of execution it is called Late Binding
- C++ - Does Early Binding when possible and Late Binding when Early Binding is not possible
- Early Binding is also known as Static Binding or Compile time Binding
- Late Binding is also known as Dynamic Binding or Runtime Binding
- For Late Binding the function being called must be present in base class as well as derived class and the call must be made using a pointer
- If call is made using an object, then it is always early bound
- Upcasted pointer - a base class pointer containing address of derived class object
- If call is made using a pointer (upcasted or not) it is late bound, if function is virtual
- To prevent an object from getting created from a class it should contain at least one pure virtual function
- A class from which an object cannot be created is called an abstract class
- If a class contains a virtual function a VTABLE is created for it. All objects of this class will have VPTR in them
- If base class contains a virtual function and the class derived from it contain a function having same prototype, then the derived class function is treated as virtual even though it is not marked virtual explicitly
- If base class contains a virtual function and the class derived from it doesn't contain a function having same prototype, then the derived class's VTABLE contains the address of the base class virtual function
- A virtual destructor ensures a proper calling order for the destructors in the class hierarchy
- In a diamond pattern inheritance creation of multiple subobjects can be prevented by marking intermediate classes as virtual