
Week 10: Abstract Class and Interfaces

The slide features decorative horizontal lines: a thick teal line at the top, a thin teal line below it, and another thick teal line at the bottom. Two short, thick brown dashes are positioned horizontally, one on the left and one on the right, centered vertically between the middle thin teal line and the bottom thick teal line.

Abstract Class

- An abstract class is a base class that will never have an object instantiated from it.
- Abstract classes are used only for inheritance, they are too general to create objects from.
- Abstract classes provide an generic base class that can be used by concrete classes to provide an interface and/or implementation.
- An abstract class in C++ is defined as any class that contains at least one pure virtual function.

Virtual Functions

- A virtual function is a member function which is declared within base class and is re-defined (Overridden) by derived class.
- When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class's version of the function.
- Virtual functions ensure that the correct function is called for an object, regardless of the type of reference (or pointer) used for function call.
- They are mainly used to achieve Runtime polymorphism
- Functions are declared with a virtual keyword in base class. The resolving of function call is done at Run-time.

Rules for Virtual Functions

- They Must be declared in public section of class.
- Virtual functions cannot be static and also cannot be a friend function of another class.
- They are always defined in base class and overridden in derived class. It is not mandatory for derived class to override (or re-define the virtual function), in that case base class version of function is used.

Virtual vs Pure Virtual

- A virtual function has an implementation in the base class; a pure virtual function does not have an implementation in the base class.
- Virtual functions can be overridden by the derived classes; pure virtual functions must be overridden by the derived classes.

Revision

- A class is abstract if it has at least one pure virtual function.
- We can have pointers and references of abstract class type.
- If we do not override the pure virtual function in derived class, then derived class also becomes abstract class.
- An abstract class can have constructors.

Interface vs Abstract Class

- An interface does not have implementation of any of its methods, it can be considered as a collection of method declarations.
- In C++, an interface can be simulated by making all methods as pure virtual.
- In Java, there is a separate keyword for interface.