MIT World Peace University

Computer Networks
Second Year B.Tech Semister 3
Academic Year 2022-23

Module 1 - Class Notes

Notes

Prepared By

P34. Krishnaraj Thadesar

Batch A2

August 30, 2022

Contents

1 Things to do

1. Types of Inheritance
2. Virtual base Classs
3. Polymorphism
4. Vitual functions

2 Inheritance

- It is the mechanism by which one class acquires the properties of another class
- Provides a way to create a new class from an existing class
- The new class is a specialized fersion of the existing class
- Inheritance establishes an "is a" relationship or a parent child relationship between classes.
- Allows sharing off the behavior of the parent class into its child classes
- child class can add new behavior or override existing behaviour from parent
- It allows a hierarchy of classes to be built moving from the most general to the most specific class.

2.1 Differnece between overloading and overiding

- Overloading is when you write the same function many times within the same class
- Overriding is when you do that same thing, but in sub classes.

2.2 Benefits of using Inheritance

- Reusablity : Reuse the methods and data of the existing class
- Extendability: Extend the existing class by adding new data and new methods.
- Modifyability: Modify the existang class by overloading its methods with newer implementations, saves memory space, increases reliability, saves the developing process.

3 Class Derivation in C++

syntax: class DerivedClassName : specification BaseClassName
like class child : public parent() // private by default ;

4 Types of Inheritance

1. Single level Inheritance : You have 1 base class –> 1 Child class.
2. Multiple Inheritance : 2 or more Base Classes –> 1 Child Class
3. Multi-Level inheritance : 1 Base Class –> 1 Child Class –> Another Child Class and so on
4. Heirarchical Inheritance : 1 Base Class –> 2 or more Child Classes.
5. Hybrid Inhertiance : Any legal combination of any of these things.

4.1 What Access modifiers mean when inheriting

1. If you do class child : private parent; then every private data member becomes inaccessible, coz anyway thats what should happen, then the protected data members become private, and public data members also become private.
2. If you do class child : protected parent; then its the same thing, except you still cant access private variables, but protected and public data members become protected
3. Same with class child : public parent; everything remains unchanged. The objects will behave in accordance with the usual laws of objects.

4.2 Constructors and Destructors in Base and Derived classes

1. Derived classes can have their own constructors and destructors
2. When an object of a derived class is created, the base class's constructor is executed frist followed by the derived class's constructor is executed first, followed bt the derived class's constructor
3. In case of multiple inheritances, the base classes are constructed in the order in which they appear in the declaration of the derived class.
4. For destructors, the order is reversed.