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 8, 2022

Contents

1 Procedural Oriented Progamming

1. Emphasis is on doing things. or Algorthms, and procedures, rather than the actual data. C++ however is the opposite.
2. Large Programs are divided into smaller programs known as functions.
3. Most of the functions share global data.
4. Data moves openly around the system from function to funnction
5. Functions transform data from one form to another. This is a problem with Security, as while transferring data, it can be poached.
6. Employs top down approach in program design.

2 Object Oriented Programming

1. Design methodology for creating a non rigid application
2. Works on entity alled as objects
3. Decompose problem in to small unit of work which are accessed via objects.
4. Emphasis is on data rather than procedure.
5. Data is hidden and cannot be accessed by external function.
6. Objects may communicate with each other through functin.
7. follows bottom up approach in program design.

2.1 Features of OOP

1. Object
2. Class
3. Data Abstraction
4. Data Encapsulation
5. Inheritence
6. Polymorphism
7. Modularity

2.2 Objects

1. Any real world entity which can have some characteristics or which can perform some work is called as Object
2. The object is called as an instance that is a copy of an entity in the programming language.
3. They are also called instances.

2.3 Class

1. A class is a plan whcih describes the object
2. It is a blue print of how the object should be represented.

3 Concepts in OOP

3.1 Data Abstraction

- Data is hidden from parts of the code and the user
- Abstraction is a rather wide topic, it doesnt just end here.
- Abstraction is sort of like the most vital thing for people working with user level software.
- You basically need to hide the useless cumbersome stuff from the programmer, and provide nice and clean methods for them to perform something, that would have been much harder if we gave them the details.
- For an example, writing data on a hard disk requires the maintainence of the speed of that disk, and where to move the head, etc. But the programmer is abstracted from this data, and only cares about creating a file.
- This concept can be extended into a single code as well, which is what we are talking about here.

3.2 Encapsulation

- It is defined as the process of enclosing one or more details from outside world through access right.
- It says how much access should be given to particular details.
- oth abstraction and Encapsulation works hand in hand because Abstraction says what details to be made visible and Encapsulation provides the level of access rights to that visible details.
- It implements the desired level of abstraction.
- Things like access modifiers fall in this category.

3.3 Inheritance

- Ability to extend the functionality from base entity to new entity belonging to same group.
- this will help us to reuse the functionality.
- We know the whole child parent example here. Stuff like Animal Class and its subclasses, or a phone class and sub classes like Samsung, and MI.
- There are 5 types of Inheritance. No one really cares about this, we just do this for segregating.
    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.

3.4 Polymorphism

- It is the ability of doing the same thing but with different type of input.
- *Many forms of single entity*
- Function overloading basically, where you are calling the same part of code, in a method, but with different names of those methods, each of which might do something unique to them, in addition to their base function.
- Compile time Polymorphism would be function overloading and operator Overloading. The Compiler figures this out.
- Runtime polymorphism : you can point to any derived class from the object of the base class at runtime that shows the ability of runtime binding. Things like virtual functions fall into this.

3.5 Object Base vs. Object Oriented Languages.

Theres not really a lot of difference here. Its just that languages like Visual basic have objects, and the concept of objects, but they dont have inhertance and dynamic Binding. C++ and Java have those features and are therefore called OOLanguages. Visual basic also has data Encapsulation, abstraction, auto initialization and clean up of objects, overloading, and is called just an Object based Language.

3.6 Applications of OOP

- Real time systems
- Simulation and modeling
- Object oriented data bases.
- Hypertext, Hyper media
- AI expect systems
- Neural networks and parallel programming.
- Decision support and office automation systems
- CIM, CAM, CAD systems.