Lecture 2
Intro. To Software Engineering and Object-Oriented Programming (2/2)

OOP

- What is problem?
  - Traditional structured programming is not sufficient for large software maintenance and reuse.
- Solution? OOP
  - Improve reusability
  - SW reliability is increased using reusable modules which are already well-tested and error-free .

OOP concepts

- Data Abstraction
  - Abstract Data Type
- Object
  - Information hiding/encapsulation
- Class
- Inheritance
- Polymorphism
- Dynamic binding

ADT(Abstract Data Type)

- A set of data variables and associated operations that are precisely specified independent of any particular implementation.
- ADT = data + operations (no implementation)
- In OOP,
  - We don't define data for defining procedures,
  - But we do define procedures necessary for defining data
  - ADT is a type that combines data and the procedures

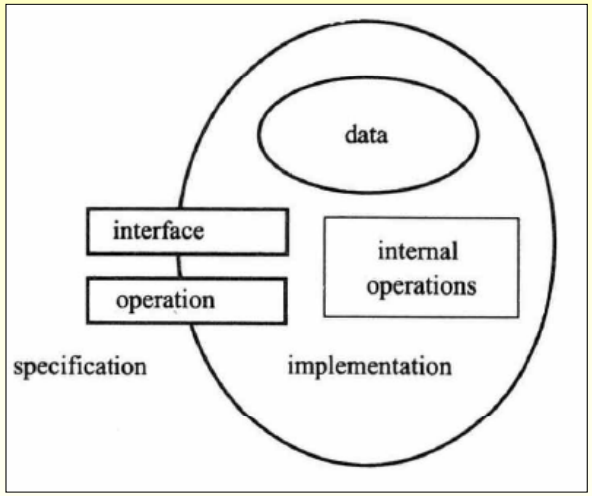
class

- In C++, “class” is used for defining ADT
- Defines attributes and behaviors of an object
  - Attributes : member variables
  - Behaviors : member functions

Object

- Object is an instance of class
- Consists of
  - Internal data
  - Internal operations
  - Interface operations
- Only class methods can access the internal data of an object.
- Message passing
  - Request some work to other object through calling a method.

Object Overview



Information Hiding/Encapsulation

- Encapsulation conceals the functional details of a class from objects that send messages to it.
- Internal data of an object can be modified only through an interface

Class vs Object

- Each object generated by the same class is different objects
  - When the objects are created, different memory space and address are assigned to each object with unique identifier.
- Objects created by the same class share the class's methods.
- An object can be dynamically constructed or destructed in memory during run-time.

Inheritance

- A child class (Sub class) inherits attributes and behaviors of parent classes (super class).
- and can define their own attributes and behaviors.
- The properties (variables,methods) of a child class should be mostly similar to the properties of parent classes.
- Goal : Object (class) Reuse
- In large SW development, code reuse reduces costs.
- We can reuse well-defined classes.
- Related programs can define new classes using inheritance.

Polymorphism

- Allows the same method (message) to different meaning.
- the ability of objects belonging to different data types to respond to method calls of the same name, each one according to an appropriate type-specific behavior.
- In C++
  - Virtual function
  - Operator overloading
  - Function overloading
  - Template

dynamic binding

- Binding
  - Determination of association between object name (identifier) and its actual memory address and value.
- Static binding
  - Binding occurs at compile-time
  - Faster compared to dynamic binding
- Dynamic binding
  - Binding occurs in run-time
  - Relatively slow
  - Sending a message to a pointer of an object. The pointer is determined in run-time and appropriate message is sent.
  - In C++, virtual function

Benefits of Object Model

- Builds a system that evolves over time
  - Entendible
  - Stable
- Thinking in terms of objects and classes is much easier for humans
- Separating the client and implementor prevents accidental damage through data encapsulation
- Reusability