C++ in depth

Classes and Objects

Saurabh Shukla (MySirG)

Agenda

1. Encapsulation
2. Implementation of Encapsulation
3. Class
4. Object
5. Example
6. Members
7. Instance member variables
8. Instance member functions

Encapsulation

An act of combining properties and methods related to the same entity is known as Encapsulation

	Person	Employee	Book
Properties	age, name	id, name, salary	bookid, title, price
Methods	setage() setName() printAge()	creatId() setName() evaluateSalary() ...	inputBook() displayBook() updateBookPrice()

Properties = Data Members, fields,
attributes, member variables

Methods = function, Procedure, action,
Service

How to implement encapsulation in C++?

There are two ways :-

1. Struct
2. Class

By default members of a structure are public and members of a class are private

Class

- Class is a keyword to create a custom data type (just like struct)
- class is a group of a variables, functions and operaters
- class is a description of an object
- class is a common noun
- class provides a blueprint for its objects
- class is a way to implement concept of encapsulation

Syntax of class

```
class  className
{
        // variables
        // functions
        // operators
};
```

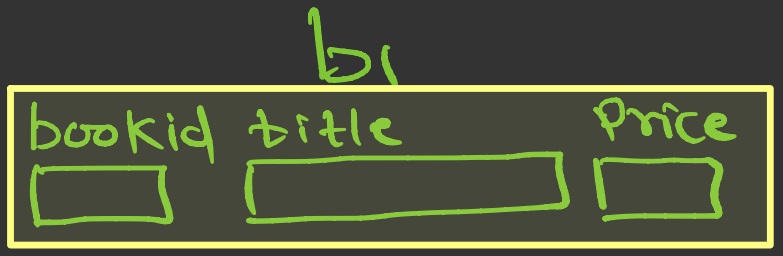
Object

- Object is a real world entity
- Object is an instance of a class
- Object is a proper noun
- Object has a state and behaviour

Object state is a set of property values at particular instant

Behaviour of an object is set of actions it can perform

Book ⟶ class



Book b1 ;

b1 is an object

b1.inputBook();
b1.displayBook();

Actions b1 can take
called behaviour

State of an object should be changed only via its methods

Define a class to represent a Complex number. Store real and imaginary part. Also define methods to set values of real and imaginary part of a complex number. Define one more method to display complex number.

$i = \sqrt{-1}$

$3 + 4i$

```
class Complex
{
    private:
        int a, b;
    public:
        void setData(int x, int y)
        {
            a=x;
            b=y;
        }
        void showData()
        {
          cout<<"a="<<a<<"b="<<b;
        }
};
```

property names

method

method

Types of variables (Scope)

- local variable
- global variable

} C, C++

- member variables — C++
  - instance member variables
  - static member variables

Members

a, b, setData() and showData() are members of Complex.

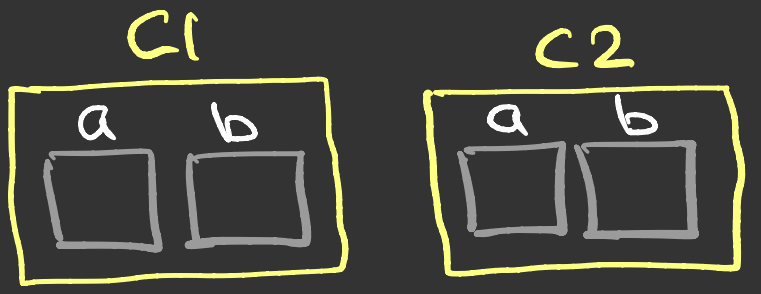
Members are of two kinds:

1. Instance members
   - Instance member variables
   - Instance member functions
2. static members
   - static member variables
   - static member functions.

Instance member variables

```
int main()
{
    Complex c1,c2;

}
```



- c1, c2 are objects
- c1 and c2 are instances of class Complex
- a and b belongs to same instance of a Complex class, therefore known as instance member variables.

Instance member function

- Functions defined in the class without static keyword are instance member functions
- Instance member function performs object specific task
- Instance member function can access any member of the current object or caller object.

- Instance member function can only be called for an object of the same class.

```
object . instanceMemberFunction()
```

Defining Instance Member functions outside the class body

```
class X
{

    function declaration

};

Returntype className :: functionname ( )
{
    ==>
}
```

Scope Resolution operator

Membership label

Instance member functions defined inside the class body are inline by default.