Objects And Classes

W2 L2

Department of Computer Science and Information Technology, University of Gujrat

Object And Classes

- Object: In OO language, we divide a program into objects rather into functions.
- Classes: A class serve as a plain or template, it specifies what data and function will be included in object of that class.

Declaring class variables

- Variables of classes (objects) are declared just like variable s of structure s and built-in data types as follows,
- **TypeName VariableName;**
- **int var; // declaring built in int data type variable**
- **Student aStudent; // declaring user defined class Student object**

Object-Orientation

- It is a kind of thinking methodology
  - Everything in the world is an object;
  - Any system is composed of objects (certainly a system is also an object);
  - The evolution and development of a system is caused by the interactions among the objects inside or outside the system

Everything in the world is an object

- A flower, a tree, an animal
- A student, a professor
- A desk, a chair, a classroom, a building
- A university, a city, a country
- The world, the universe
- A subject such as CS, IS, Math, History, …

Any system is composed of objects

- A law system
- A cultural system
- An educational system
- An economic system
- An Information system
- A computer system

Question ?

- To design a class, we should define attributes and behaviors.
  - Student class
  - Computer class
  - Customer class

Example

- Let's design a class for **ShoppingCart**:
  - The class Shopping Cart would consist of
    - features shared by all shopping cart, such as
      - Size,
      - Owner and
      - Quantities of products (characteristics),
    - And the ability to
      - Check whether it is full,
      - Calculate total price,
      - Add a product,
      - Delete a product (behavior).

A C++ Class

- In C++:
  - Data members: variables which are used to represent the attributes of a class.
  - Member functions: functions which are used to define the behaviors of a class.
- A C++ class has two parts:
  - Interface of a Class
  - Implementation of a Class

The Interface

- The interface of a Class like a ***manual***
  - Describes what the objects of this class can do for us, and also how to request these services from objects.
  - Allow compiler to recognize the classes when used elsewhere.
- The interface includes
  - Data Members
  - Member Function Prototypes

Example: stuent Class

- Data Member: `int student_name;`
- `Data Member: int rNoll;`
- Member function: `void getstudentdata();`
- `class studentRecord`

```
{
  public: void displayMessage();
  private: int courseNumber;
};
```

Accessing members

- Members of an object can be accessed using,
  - dot operator (.) to access via the variable name
  - Student aStudent; // declaring Student object
  - **aStudent. rollNo = 5;**
- arrow operator (->) to access via a pointer to an object
  - Student * aStudent = new Student();
  - **// declaring and initializing Student pointer**
  - **aStudent->rollNo = 5;**
- Note: it is against the principle of OOP to access the data members directly using object of class as we have done above. This code is given for example only we should write assessor functions (setters and getters) wherever we want to access themembers of the class.
- **Member functions are accessed in the similar way using dot or arrow operator.**

Access specifiers

- These are used to enforce access restrictions to members of a class, there are three access specifiers,
- **'public'**
- is used to tell that member can be accessed whenever you have access to the obj
- **'private'**
- is used to tell that member can only be accessed from a member function
- **'protected'**
- to be discussed when we cover inheritance

C++ Class – Example

```
1 class Time {
2 public:
3    Time();
4    void setTime( int, int, int );
5    void printMilitary();
6    void printStandard();
7 private:
8    int hour;     // 0 - 23
9    int minute;   // 0 - 59
10   int second;   // 0 - 59
11};
```

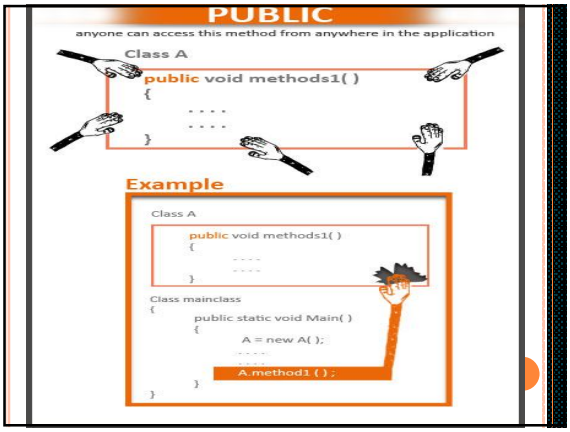
Public: and Private: are member-access specifiers.

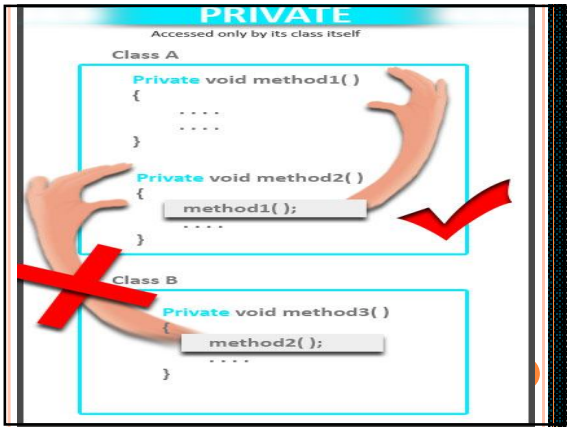
setTime, printMilitary, and printStandard are member functions. Time is the constructor.

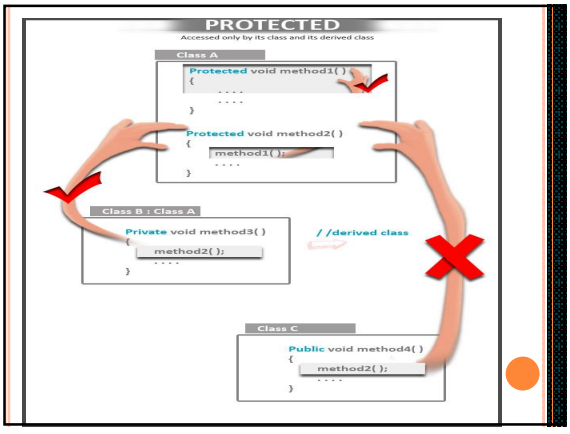
hour, minute, and second are data members.

PUBLIC: OR PRIVATE:

- It is not compulsory for all members functions to be public and all data members to be private.
- Some member functions can also be private.
- Similarly, data members can also be public.
  - Violation of *Encapsulation* principle







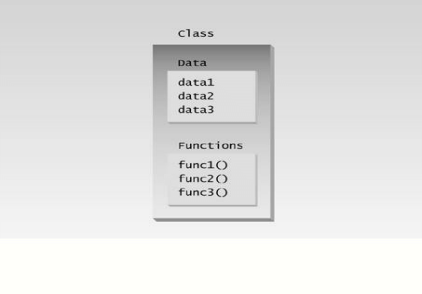
A Simple Class (1)

```
#include <iostream>
using namespace std;
class smallobj              //declare a class
{
private:
int somedata;          //class data
public:
void setdata(int d)    //member function to set data
{ somedata = d; }
void showdata()        //member function to display data
{ cout << "Data is " << somedata << endl; }
};
```

A Simple Class (2)

```
int main()
{
smallobj s1, s2;   //define two objects of class smallobj
s1.setdata(1066);  //call member function to set data
s2.setdata(1776);
s1.showdata();     //call member function to display data
s2.showdata();
return 0;
}
```

A Simple Class (3)



Class and Objects

- Object is an instantiation of a user defined type or class. Once we have defined a class we can create as many objects for that class as we require.
- We *define* two objects s1 and s2 that are instances of that class.

Declaring the Class

```
class smallobj              //declare a class
{
private:
int somedata;          //class data
public:
void setdata(int d)    //member function to set data
{ somedata = d; }
void showdata()        //member function to display data
{ cout << "\nData is " << somedata; }
};
```

Declaring the Class (1)

- declaration starts with the keyword **class.**
- Like a structure, the body of the class is delimited by braces and terminated by a semicolon.

PRIVATE AND PUBLIC

- Body of the class contains two unfamiliar keywords private and public.
- key feature of object–oriented programming is *data hiding*
- Private not accessible from outside class.
- Public accessible from outside class.
- Data hiding with the security techniques used to protect computer databases.

CLASS DATA

- The data items within a class smallobj somedata int are called *data members* (or sometimes *member data*).
- There can be any number of data members in a class, just as there can be any number of data items in a structure.

FUNCTIONS ARE PUBLIC, DATA IS PRIVATE

- Usually the data within a class is private and the functions are public.
- There is no rule that data must be private and functions public; in some circumstances you may find you'll need to use private functions and public data.

MEMBER FUNCTIONS WITHIN CLASS DECLARATION

- Member functions defined inside a class this way are created as inline functions by default.
- Functions defined outside the class are not normally inline.

INITIALIZATION

- Class data member cannot be initialized at declaration time.
- By default all data members have garbage value.
- If all members are declared public then, they can be initialized at object declaration time.
  - Similar to structure initialization time.
- Otherwise we require a constructor for this.

DEFINING OBJECTS

- Defining an object is similar to defining a variable of any data type: Space is set aside for it in memory.
- This is also called *instantiating* them. The term *instantiating* arises because an *instance* of the class is created.
- An object is an instance (that is, a specific example) of a class. Objects are sometimes called *instance variables*

.