

C++ Programming

Trainer : Rajiv K

Email: rajiv.kamune@sunbeaminfo.com



Agenda

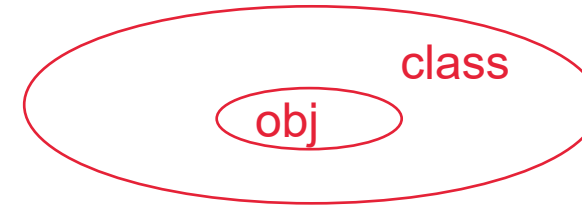
- class and object
- Data Member & Member Functions
- Types of Member Functions
- Constructor
- Facilitator
- Inspector and Mutator
- Destructor
- this pointer
- Constructor initializer list




class and object

- Everything in C++ is associated with **classes and objects**, along with its properties and behavior.

- The class is **template** for an **Object**



- Or you may say it is a **blue print** for an **object**, by looking towards which an object gets **instantiated**.
- Class is collection of **data member** and **member function**.
-  Object is an **instance** of a class.
- Entity that has **physical existence**, **can store data**, **send and receive message** to communicate with other objects.

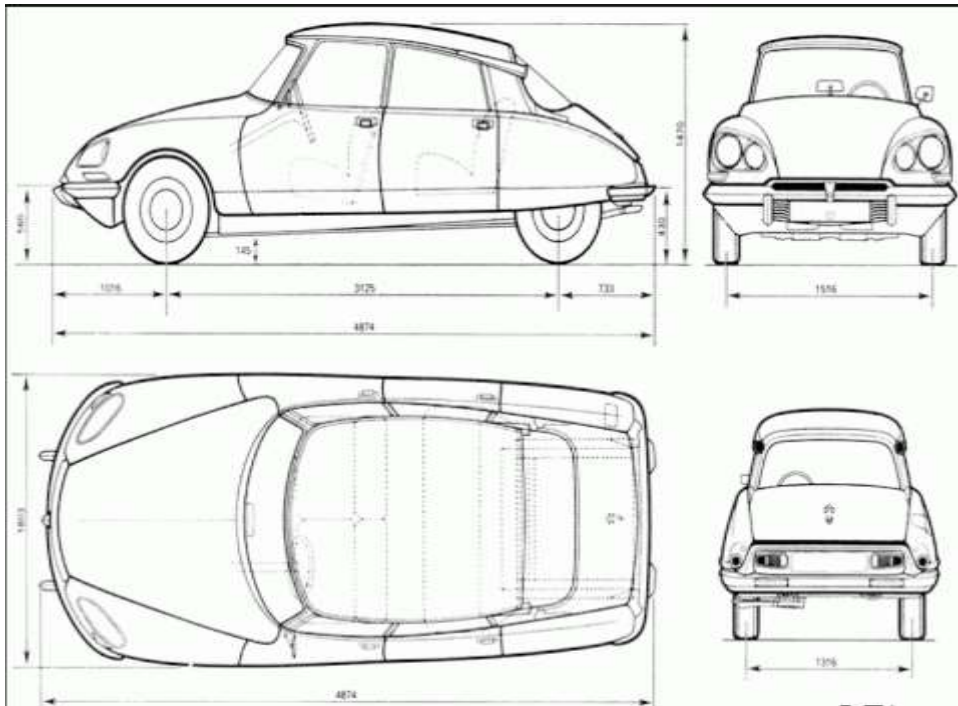


Example of Class and Object

CLASS :- CAR

OBJECT :- SWIFT,AUDI ETC

Blue Print of Car



Objects of a
Class Car



class and object

- Example:

Blue Print of building



Real life entity(Object)





Object

- An entity, which **get space inside memory** is called object.
- Object is used to access data members and member function of the class
- **Process of creating object from a class is called instantiation**
- **Object has**
 - Data members (**state** of object) **PROPERTIES**
 - Value stored inside object is called **state of the object**.
 - Value of data member represent **state of the object**.
 - Member function (**behavior** of object)
 - Set of operation that we perform on object is called behaviour of an object.
 - Member function of class represent behaviour of the object.
 - is how object **acts & reacts**, when its state is changed & operations are done
 - **Operations performed are also known as messages**

Point p1
HERE Point is class
and p1 is an object.



Object

- **State:** State of an object contains all the **properties** of the object and **values of** each and every the **properties**. 
- **Value Stored in side an object** is called state of an object.
- **Behavior:** Behavior is how an object **act and react**, when state of an object changes and function gets called. 
- **Member function** of a class is nothing but the **behavior** of an object.
- **Identity:** Identity is the property of a class by which it gets **differentiated** from all other objects.



Data Member & Member Functions

- Unique address(***identity*** of object)
 - Value of any data member, which is used to identify object uniquely is called its identity.
 - If state of object is same then its **address** can be considered as its identity.
- **Data members are the data variables and member functions are the functions used to manipulate these variables** and together these data members and member functions defines the properties and behavior of the objects in a Class.
- The data members and member functions of class can be accessed using the dot('.')

syntax:-

```
class <class name>
{
    data members;

    member functions;

};
```



Types of Member Functions

- Constructor (TO INITIALIZE)
- Facilitator
- Inspector (GETTER)
- Mutator (SETTER)
- Destructor (TO DEINITIALIZE)
- **Constructor:**
 - Constructor is a **special member** function of a class **having same name** as a class and **having no return type**. *since we only initialize an object using it and we don't perform any operation therefore no need of return type.*
 - Constructor is used to **initialize an object/data member** of a class.
 - In entire **life cycle** of an object constructor **gets called only once**.

```
class CAR
{
    CAR( )
    {
    }
}
```

We are not supposed to call constructor the moment we declare the object of a class at that moment constructor is called.



Types of Constructor

- **Types of Constructor:**
 - 1. Parameterless Constructor
 - 2. Parameterised Constructor
 - 3. Default Constructor
- **1. Parameterless Constructor:**
 - Parameterless constructor is a constructor which **does not have parameter** is called as parameterless constructor.
 - It is also called as **zero parameter constructor** of user defined default constructor.
- **2. Parameterised Constructor:**
 - Parameterised constructor is a constructor which **dose have one or more parameter** is called as parameterless constructor.
- **3. Default Constructor:**
 - If no constructors are explicitly declared in the class, a default constructor is **provided automatically by the compiler**



Types of Member Functions

- **Facilitator:**
 - Cause an object to perform some action or service.
- **Inspector:** (GETTER) can return private data member outside the class.
 - public member functions in a class that get (accessors) the value of data member of a class.
- **Mutator:** (SETTER)
 - a mutator method is a method used to **change the value** of a data member of the class.

INSPECTOR:

```
int getX()  
{  
    return x;  
}
```

MUTATOR:

```
void setX(int a)  
{  
    x=a;//assignment  
}
```



Types of Member Functions

- **Destructor:**

- Destructor is also a special member function like constructor.
- An operation that **deinitializes an object**.
- A member function that is **invoked automatically** when the **object goes out of scope or is** explicitly destroyed by a call to **delete**.
- Destructor neither **requires any argument nor returns any value**.



This Pointer this->

- this is a key word in C++
- It is **implicit constant pointer** available inside each non static member function of a class.
- This pointer stores the address of **current object** of a class.
- If name of data member and parameter passed to the member function is same then it becomes **mandatory** to use **this pointer** inside that function to differentiate between the data member and local parameter to the function.
- Following are the function which **doesn't** get this pointer
 - Global Function
 - Static Function
 - Friend Function

THIS pointer will store address of object



Constructor's member initializer list

- If we want to initialize data members according to users requirement then we should use constructor body.

```
class Test
{
private:
    int num1;
    int num2;
    int num3;

public:
    Test( void )
    {
        this->num1 = 10;
        this->num2 = 20;
        this->num3 = num2;
    }
};
```

- If we want to initialize data member according to order of data member declaration then we can use **constructors member initializer** list.

```
class Test
{
private:
    int num1;
    int num2;
    int num3;

public:
    Test( void ) : num1( 10 ), num2( 20 ), num3( num2 )
    { }
};
```

Except array we can initialize any member inside constructors member initializer list.



Empty Class

Empty Class :-

A class which **do not have any Data member or member function** is called as the Empty class.

- We **can create an object** of a empty class.
- **Size** of an object of a empty class is **1 byte**.



Difference between class and Structure in C++

Class

Members of a class are **private** by default.

Member classes/structures of a class are private by default.

It is declared using the **class** keyword.

It is normally used for **data abstraction** and further **inheritance**.

Structure

Members of a structure are **public** by default.

Member classes/structures of a structure are public by default.

It is declared using the **struct** keyword.

It is normally used for the **grouping of data**



Thank You

