### Object Oriented Programming in C++

#### Why Object-Oriented Programming ???

Before we discuss object-oriented programming, we need to learn why we need object-oriented programming?

* C++ language was designed with the main intention of adding object-oriented programming to C language
* As the size of the program increases readability, maintainability, and bug-free nature of the program decrease.
* This was the major problem with languages like C which relied upon functions or procedure (hence the name procedural programming language)
* As a result, the possibility of not addressing the problem adequately was high
* Also, data was almost neglected, data security was easily compromised
* Using classes solves this problem by modeling program as a real-world scenario

#### Difference between Procedure Oriented Programming and Object-Oriented Programming:

##### **1.Procedure Oriented Programming:**

* Consists of writing a set of instruction for the computer to follow
* The main focus is on functions and not on the flow of data
* Functions can either use local or global data
* Data moves openly from function to function

##### **2.Object-Oriented Programming:**

* Works on the concept of classes and object
* A class is a template to create objects
* Treats data as a critical element
* Decomposes the problem in objects and builds data and functions around the objects

#### Basic Concepts in Object-Oriented Programming:

* **Classes -**Basic template for creating objects.
* **Objects –**Basic run-time entities
* **Data Abstraction & Encapsulation –**Wrapping data and functions into a single unit
* **Inheritance –**Properties of one class can be inherited into others
* **Polymorphism –**Ability to take more than one forms
* **Dynamic Binding –**Code which will execute is not known until the program runs
* **Message Passing –**message (Information) call format

#### Benefits of Object-Oriented Programming:

* Better code reusability using objects and inheritance
* Principle of data hiding helps build secure systems
* Multiple Objects can co-exist without any interference
* Software complexity can be easily managed

### Classes, Public and Private access modifiers in C++

#### Why use classes instead of structures ???

Classes and structures are somewhat the same but still, they have some differences. For example, we cannot hide data in structures which means that everything is public and can be accessed easily which is a major drawback of the structure because structures cannot be used where data security is a major concern. Another drawback of structures is that we cannot add functions in it.

#### Classes in C++:

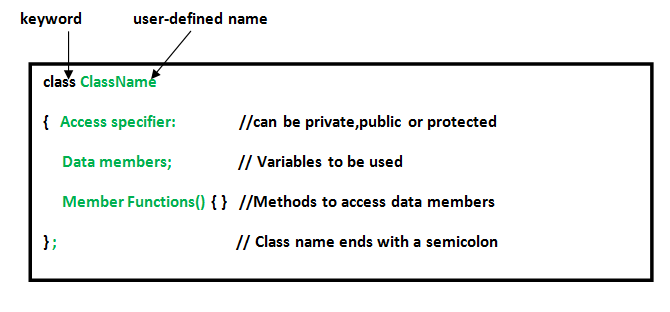
Classes are user-defined data-types and are a template for creating objects. Classes consist of variables and functions which are also called class members.

**Class:** **A class in C++ is the building block, that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.**For Example: Consider the Class of **Cars**. There may be many cars with different names and brand but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range etc. So here, Car is the class and wheels, speed limits, mileage are their properties.

* A Class is a user defined data-type which has data members and member functions.
* 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 behaviour of the objects in a Class.
* In the above example of class Car, the data member will be speed limit, mileage etc and member functions can be apply brakes, increase speed etc.

**An Object** **is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.**

**Defining Class and Declaring Objects:**

A class is defined in C++ using keyword class followed by the name of class. The body of class is defined inside the curly brackets and terminated by a semicolon at the end.

**Declaring Objects:** When a class is defined, only the specification for the object is defined; no memory or storage is allocated. To use the data and access functions defined in the class, you need to create objects.

**Syntax:**

**ClassName ObjectName;**

**Accessing data members and member functions**: The data members and member functions of class can be accessed using the dot(‘.’) operator with the object. For example if the name of object is obj and you want to access the member function with the name printName() then you will have to write obj.printName() .

**Accessing Data Members**

The public data members are also accessed in the same way given however the private data members are not allowed to be accessed directly by the object. Accessing a data member depends solely on the access control of that data member.  
This access control is given by [Access modifiers in C++](https://www.geeksforgeeks.org/access-modifiers-in-c/). There are three access modifiers : **public, private and protected**.

There are 3 types of access modifiers available in C++:

1. **Public**
2. **Private**
3. **Protected**

**Note**: **If we do not specify any access modifiers for the members inside the class then by default the access modifier for the members will be Private.**

Let us now look at each one these access modifiers in details:

 **1. Public**: All the class members declared under the public specifier will be available to everyone. The data members and member functions declared as public can be accessed by other classes and functions too. The public members of a class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

**2.Private**: The class members declared as *private* can be accessed only by the member functions inside the class. They are not allowed to be accessed directly by any object or function outside the class. Only the member functions or the [friend functions](https://www.geeksforgeeks.org/friend-class-function-cpp/) are allowed to access the private data members of a class.

**3. Protected**: Protected access modifier is similar to private access modifier in the sense that it can’t be accessed outside of it’s class unless with the help of friend class, the difference is that the class members declared as Protected can be accessed by any subclass(derived class) of that class as well.

**Note**: This access through inheritance can alter the access modifier of the elements of base class in derived class depending on the [modes of Inheritance](https://www.geeksforgeeks.org/inheritance-in-c/#Modes%20of%20Inheritance).

#### Code as described/written in the video

#include<iostream>

using namespace std;

class Employee

{

private:

int a, b, c;

public:

int d, e;

void setData(int a1, int b1, int c1); // Declaration

void getData(){

cout<<"The value of a is "<<a<<endl;

cout<<"The value of b is "<<b<<endl;

cout<<"The value of c is "<<c<<endl;

cout<<"The value of d is "<<d<<endl;

cout<<"The value of e is "<<e<<endl;

}

};

void Employee :: setData(int a1, int b1, int c1){

a = a1;

b = b1;

c = c1;

}

int main(){

Employee harry;

// harry.a = 134; -->This will throw error as a is private

harry.d = 34;

harry.e = 89;

harry.setData(1,2,4);

harry.getData();

return 0;}