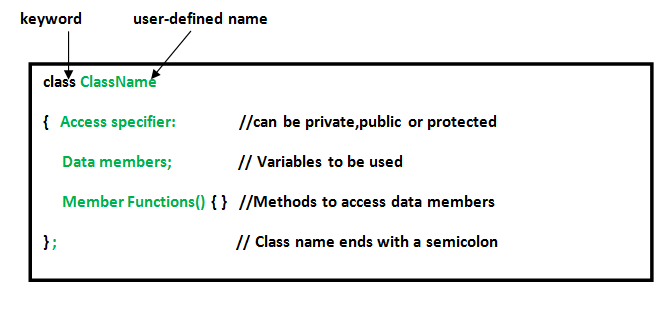
## C++ Classes/Objects

C++ is an object-oriented programming language.

Everything in C++ is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

Attributes and methods are basically variables and functions that belongs to the class. These are often referred to as "class members".

A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.



class MyClass { // The class

public: // Access specifier

int myNum; // Attribute (int variable)

string myString; // Attribute (string variable)

};

### Example

class MyClass { // The class

public: // Access specifier

int myNum; // Attribute (int variable)

string myString; // Attribute (string variable)

};

int main() {

MyClass myObj; // Create an object of MyClass

// Access attributes and set values

myObj.myNum = 15;

myObj.myString = "Some text";

// Print attribute values

cout << myObj.myNum << "\n";

cout << myObj.myString;

return 0;

}

## Multiple Objects

// Create a Car class with some attributes

class Car {

public:

string brand;

string model;

int year;

};

int main() {

// Create an object of Car

Car carObj1;

carObj1.brand = "BMW";

carObj1.model = "X5";

carObj1.year = 1999;

// Create another object of Car

Car carObj2;

carObj2.brand = "Ford";

carObj2.model = "Mustang";

carObj2.year = 1969;

// Print attribute values

cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";

cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";

return 0;

}

**Member Functions in Classes**

There are 2 ways to define a member function:

* Inside class definition
* Outside class definition

To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

// C++ program to demonstrate function

// declaration outside class

#include <bits/stdc++.h>

using namespace std;

class Geeks

{

public:

string geekname;

int id;

// printname is not defined inside class definition

void printname();

// printid is defined inside class definition

void printid()

{

cout << "Geek id is: " << id;

}

};

// Definition of printname using scope resolution operator ::

void Geeks::printname()

{

cout << "Geekname is: " << geekname;

}

int main() {

Geeks obj1;

obj1.geekname = "xyz";

obj1.id=15;

// call printname()

obj1.printname();

cout << endl;

// call printid()

obj1.printid();

return 0;

}