C++ Classes and Objects

Dr. Anish Soni

Introduction

- **In C++, classes and objects are the basic building** block that leads to Object-Oriented programming in C++
- **A class 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 brands but all of them will share some common properties like all of them will have *4 wheels*, *Speed Limit*, *Mileage range,* etc.
- So here, the Car is the class, and wheels, speed limits, and mileage are their properties.

- A Class is a user-defined data type that has data members and member functions.

- Data members are the data variables and member functions are the functions used to manipulate these variables together.

- These data members and member functions define 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 *applying brakes*, *increasing speed,* etc.

But we cannot use the class as it is. We first have to create an object of the class to use its features. An Object is an instance of a Class.

- ***Note:*** *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 in C++

- A class is defined in C++ using the keyword c**lass** followed by the name of the class. The following is the syntax:

class ClassName

{

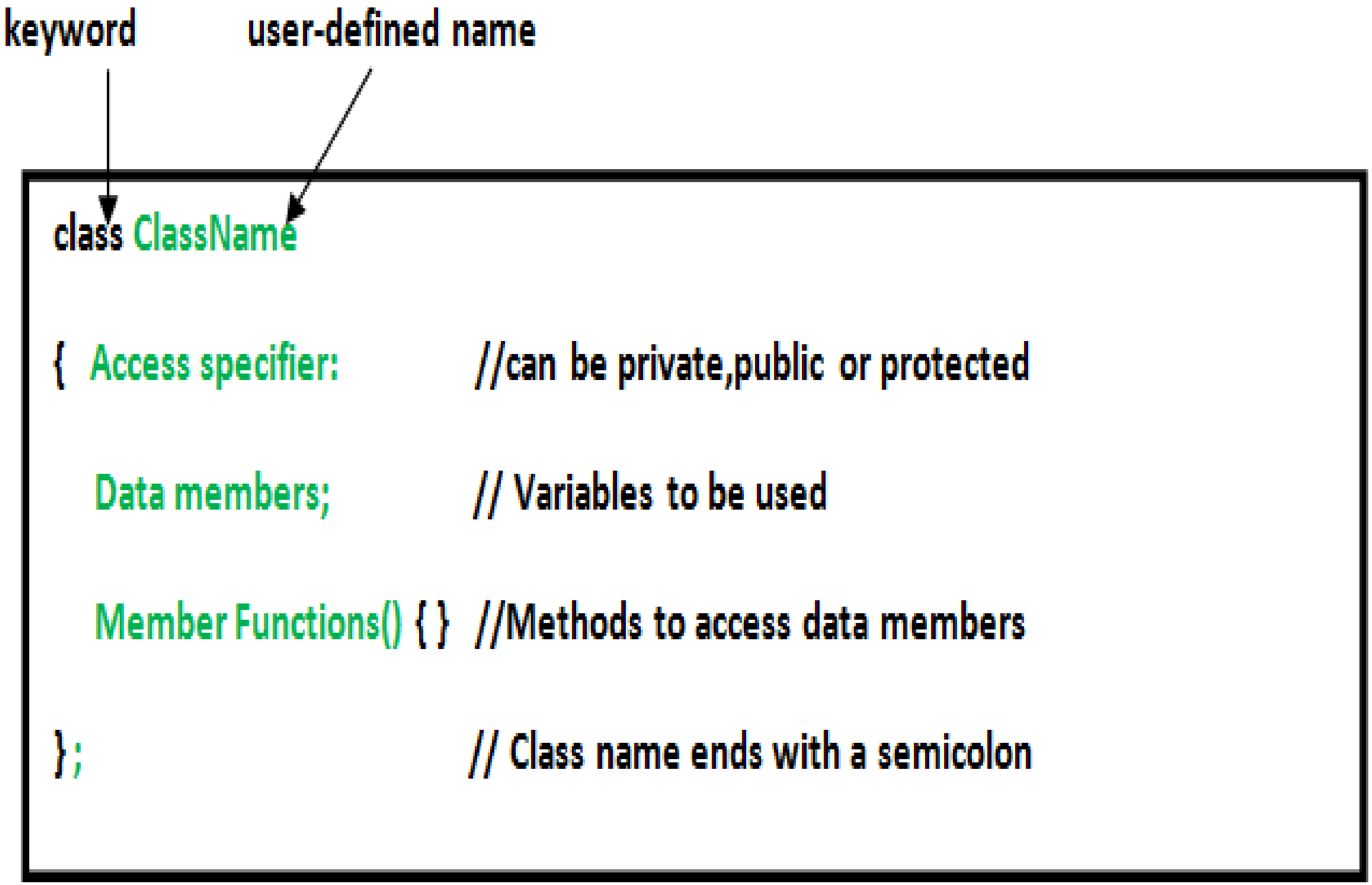
access_specifier:

// Body of the class

};

Here, the access specifier defines the level of access to the class's data members.

- **Example**
- class ThisClass
- {
- public:
- int var; *// data member*
- void print() *// member method*
- {
- cout << "Hello";
- }
- };



```
class ClassName

{  Access specifier:          //can be private,public or protected

   Data members;              // Variables to be used

   Member Functions() { }     //Methods to access data members

};                            // Class name ends with a semicolon
```

What is an Object in C++?

- 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 to Create an Object

- We can create an object of the given class in the same way we declare the variables of any other inbuilt data type.

  ClassName ObjectName;

Example

MyClass obj;

In the above statement, the object of MyClass with name obj is created.

Accessing Data Members and Member Functions

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

  obj.printName()

Example of Class and Object in C++

```
// C++ program to illustrate how create a simple class and // object
 #include <iostream.h>
#include <string.h>
class Person       // Define a class named 'Person'
{
public:
string name;       // Data member;
int age;             // Data member;

void introduce()     //Member function to introduce the person
{
cout << "Hi, my name is " << name << " and I am " << age << " years old." << endl;
}

 };

int main()
 {
Person person1;                  // Create an object of the Person class

person1.name = "Alice";    // accessing data members

 person1.age = 30;            // accessing data members

  person1.introduce();        // Call the introduce member method
return 0;
}
```

Access Modifiers

- In C++ classes, we can control the access to the members of the class using Access Specifiers. Also known as access modifier.
- They are the keywords that are specified in the class and all the members of the class under that access specifier will have particular access level.
- In C++, there are 3 access specifiers that are as follows:
- **Public:** Members declared as public can be accessed from outside the class.
- **Private:** Members declared as private can only be accessed within the class itself.
- **Protected:** Members declared as protected can be accessed within the class and by derived classes.

  If we do not specify the access specifier, the private specifier is applied to every member by default.