

CHRISTIAN EMINENT COLLEGE, INDORE
(Academy of Management, Professional Education and Research)
An Autonomous Institution Accredited with 'A' Grade by NAAC



E-Content

On

“Class and Access Modifier in C++”

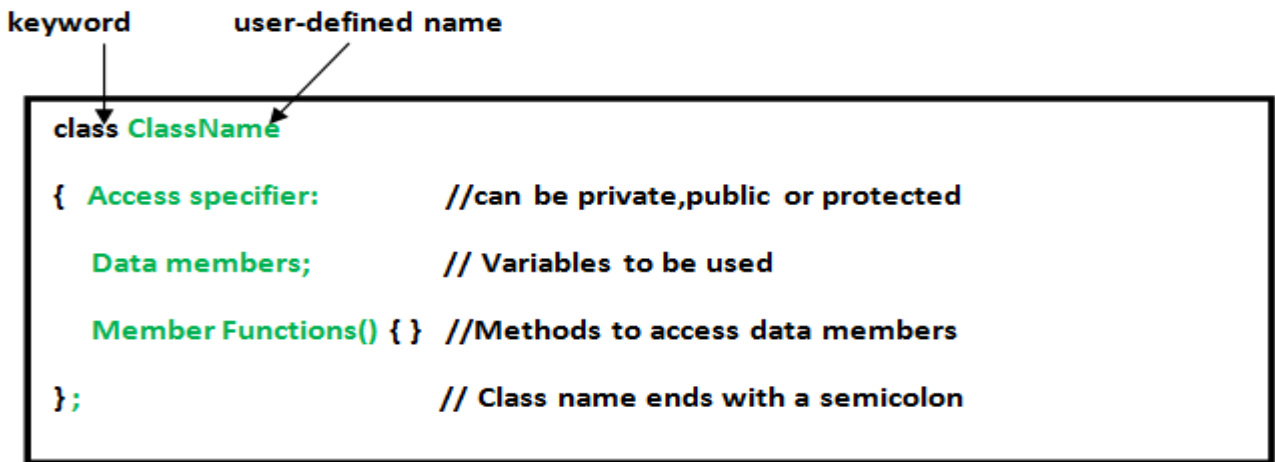
Prepared By: Prof. Amit Kumar Nilosey

Department of Computer Science & Electronics

Class and Object

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.



Object:

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.

Simple class Example

```
class item
{
    int number;
    float cost;
    public:
    void getdata(int a , float b);
    void putdata ();
};
```

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:

Class name object name

Example:

Item itm;

Simple class Program

```
#include<iostream.h>

#include<conio.h>

class Addition
{
private:           //Access - Specifier
int num1, num2, result;   //Member Variable Declaration
public:          //Access - Specifier
void read()
{
cout<<"Enter Number 1 :";
cin>> num1;
cout<<"Enter Number 2 :";
cin>> num2;
}
void sum()
{
    result = num1 + num2;
}
void print()
{
cout<<"Result :"<< num1 <<" + "<< num1 <<" = "<< result <<endl;
}
};
```

```

void main()
{
    Additions ad;
    ad.read();
    ad.sum();
    ad.print();
    getch();
}

```

Output:

Enter Number1 :100

Enter Number2 :200

Result :100 + 100 = 300

C++ Class Access Modifiers

Data hiding is one of the important features of Object Oriented Programming which allows preventing the functions of a program to access directly the internal representation of a class type. Access modifiers in C++ class define the access control rules. C++ has 3 new keywords introduced, namely,

1. public
2. private
3. protected

Public Access Modifier

A public member is accessible from anywhere outside the class but within a program. You can set and get the value of public variables without any member function

```

class Line
{
    public:
    double length;
    void setLength( double len );
    double getLength( void );
};

```

Private Access Modifier

A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members.

By default all the members of a class would be private, for example in the following class width is a private member, which means until you label a member, it will be assumed a private member –

```
class Box
{
double width;
public:
double length;
void setwidth( double wid );
};
```

Protected Access Modifier

A **protected** member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.

For now you can check following example where I have derived one child class **SmallBox** from a parent class **Box**.

```
class Box
{
protected:
double width;
};

class SmallBox:Box
{
// SmallBox is the derived class.
public:
void setSmallWidth( double wid );
double getSmallWidth( void );
};
```

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.

Function definition Inside the class

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class item
{
int code,price;
char name[20];
public:
void getdata(int a , int b , char n[])
{
code=a;
price=b;
strcpy(name,n);
}
void display()
{
cout<<code;
cout<<price;
cout<<name;
}
};
void main()
{
item x;
int a,b;
char c[20];
```

```
cout<<"enter the value of a ";
cin>>a;
cout<<"enter the value of b ";
cin>>b;
cout<<"enter the name ";
cin>>c;
x.getdata(a,b,c);
x.display();
getch();
}
```

Function Definition Outside the class

```
#include<iostream.h>
#include<conio.h>
class large
{
int a,b;
void largest();
public:
void getdata()
{
cout<<"enter the value";
cin>>a ;
cout<<"enter the value of b";
cin>>b;
}
void display()
{
largest();
}
};
```

```
void item :: largest() // Function Definition outside the class
{
if(a>b)
cout<<"a is greater";
else
cout<<"b is greater";
}
void main()
{
large x;
x.getdata();
x.display();
getch();
}
```