Class & Objects

The mechanism that allows you to combine data and the function in a single unit is called a class. Once a class is defined, you can declare variables of that type. A class variable is called object or instance. In other words, a class would be the data type, and an object would be the variable. Classes are generally declared using the keyword class, with the following format:

```
class class_name
{
    private:
        members1;
    protected:
        members2;
    public:
        members3;
};
```

Where class_name is a valid identifier for the class. The body of the declaration can contain members, that can be either data or function declarations, The members of a class are classified into three categories: private, public, and protected. private, protected, and public are reserved words and are called member access specifiers. These specifiers modify the access rights that the members following them acquire.

private members of a class are accessible only from within other members of the same class. You cannot access it outside of the class.
protected members are accessible from members of their same class and also from members of their derived classes.
Finally, **public members** are accessible from anywhere where the object is visible.

By default, all members of a class declared with the class keyword have private access for all its members. Therefore, any member that is declared before one other class specifier automatically has private access.

Here is a complete example :

```
class Circle
{
    private:
        double radius;
    public:
        void setRadius(double r)
        {
            radius = r;
        }
        double getArea()
        {
            return 3.14 * radius * radius;
        }
};
```

Object Declaration

Once a class is defined, you can declare objects of that type. The syntax for declaring a object is the same as that for declaring any other variable. The following statements declare two objects of type circle:

```
Circle c1, c2;
```

Accessing Class Members

Once an object of a class is declared, it can access the public members of the class.

```
c1.setRadius(2.5);
```

Defining Member function of class

You can define Functions inside the class as shown in above example. Member functions defined inside a class this way are created as inline functions by default. It is also possible to declare a function within a class but define it elsewhere. Functions defined outside the class are not normally inline.
When we define a function outside the class we cannot reference them (directly) outside of the class. In order to reference these, we use the scope resolution operator, :: (double colon). In this example, we are defining function setRadius outside the class:

```
void Circle :: setRadius(double r)
{
    radius = r;
}
```

The following program demonstrates the general feature of classes. Member functions setRadius() and getArea() defined outside the class.

```
#include <iostream>
using namespace std;

class Circle //specify a class
{
    private :
        double radius; //class data members
    public:
        void setRadius(double r);
        double getArea(); //member function to return area
};

void Circle :: setRadius(double r)
{
    radius = r;
}

double Circle :: getArea()
{
    return 3.14 * radius * radius;
}

int main()
{
    Circle c1; //define object of class circle
    c1.setRadius(2.5); //call member function to initialize radius
    cout << c1.getArea(); //display area of circle object
    return 0;
}
```