CLASSES AND OBJECTS

Class:

Collection of objects sharing common properties & behavior. Or Simply Collection of similar objects.

Need:

To define real world objects more effectively.(i.e. not only data associated with them but also their associated behavior or operations.

Defining Classes:

Declaration of Class

1. Data Members
2. Member Functions
3. Program access levels
4. Class Tagname

Syntax

```
class < Class Name >
{
private :
[Var Declaration]
[Function Declarations]
protected :
[Var Declaration]
[Function Declarations]
public :
[Var Declaration]
[Function Declarations]
};
```

Example :

```
class A
{ int a;
public:
void getdata()
{
cin>>a;
}
void showdata()
{
cout<<a;
}
};
```

Note: Functions defined within the scope of the class are by default inline. No need to use inline keyword with them to make them inline.

Access specifiers:

Access specifiers are used to identify access rights for the data and member functions of the class. There are three main types of access specifiers in C++ programming language:

- private
- public
- protected

Private:
A *private* member within a class denotes that only members of the same class have accessibility. The *private* member is inaccessible from outside the class.
(Note: Default access specifier, if no access specifier is provided then all the Data members and member functions are treated as private members).

Public:
Public members are accessible from outside the class.

Protected:
A protected access specifier is a stage between *private* and *public* access. If member functions defined in a class are *protected*, they cannot be accessed from outside the class but can be accessed from the derived class.
While defining access specifiers, the programmer must use the keywords: *private*, *public* or *protected* when needed, followed by a semicolon and then define the data and member functions under it.

Defining function outside the class (out of scope of class)
Member Functions of a class can be defined outside the class. For this purpose the scope resolution operator (: :) can be used.

```
Return_type class_name : : function_name( argument list)
{
// statements to be executed
}
```

Example

```
class A
{
int a;
Public:
void getdata();
void showdata();
};
void A:: getdata()
{
Cin>>a;
}
void A:: showdata()
{
Cout<<a;
}
```

Use of a Class in a Program
Define the following in a sequence to use a class in a program
1. **Class Definition**

2. **Class Method**
3. **In main( ) Create Object**

Example:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class stud
{private: //private members
int sno;
char sname[80];
int cls;
public: //public members
void input(void);
void output(void);
};
void stud :: input(void)
{
cout<<"Enter Sno : ";
cin>>sno;
cout<<"Enter Sname : ";
gets(sname);
cout<<"\nEnter Class :";
cin>>cls;
}
void stud :: output(void)
{
cout<<"\n Sno : "<<sno;
cout<<"\n Sname : "<<sname;
cout<<"\n Class : "<<cls;
}
void main()
{ clrscr();
stud s;
s.input();
s.output();
getch();
}
```

Important Points:

- The private members provide data hiding by preventing us from accessing the data directly
- The class declaration must end with a semi-colon.
- It is important to include the corresponding *.h* file; otherwise, we will get compile-time errors. In the #*include*, the name of the file is enclosed in quotes, not in angle brackets. Angle brackets are used for including standard library header files, and quotes are used for including our own header files (this tells the compiler where to look for the header file).

1 What do you understand about a member function? How does a member function differ from an ordinary function?

Ans: a member function is the set of operations that are applied to objects of that class. A class may have zero or more member functions. they are also referred as class interface. An ordinary function is set of statement enclosed within curly braces and can be accessed by an ordinary variable. No security is provided in an ordinary function.

OR

A member function is a function that can be defined either inside or outside the class definition and is a part of class and can not be accessible out side of the class without class object. But ordinary function is just like a normal function and can be access anywhere in the program freely.

Creation of Objects:

Once the class is created, one or more objects can be created from the class as objects are instance of the class. Just as we declare a variable of data type *int* as:

```
int x;
```

Objects are also declared as:

```
class name followed by object name;
ex e1;
```

This declares *e1* to be an object of class ex .

For example a complete class and object declaration is given below:

```
class ex
{ private: int x,y; public:
```

ASSIGNMENT

Define a class candidate in C++ with following Description:

Private Members

- A data member RNo (Registration Number) of type long
- A data member Name of type string
- A data member Score of type float
- A data member Remark of type string
- A member function AssignRem( ) to assign Remarks as per the Score obtained by a candidate.

Score range and the respective Remarks are shown as follows:

Score	Remarks
>=50	Selected
less than 50	Not selected

Public members

- A function ENTER( ) to allow user to enter values for RNo, Name, Score & call function AssignRem() to assign the remarks.
- A function DISPLAY( ) to allow user to view the content of all the data members.

↘↘↘ ↓