Date

Creating the objects of a class

A class is a user defined data type, while an object is an instance of a class. The objects of a class are declared after the class definition. One create several objects from a class. Objects occupy spaces in memory according to the class definition. The data members of a class occupy memory space inside the objects of that class, while member function doesnot occupy any space inside the objects.

The general form for creating object is

```
class-name   object 1, object 2, ....., object n;
```

eg.   student S1, S2, S3;

Here S1, S2 and S3 are the objects created from class student. The objects are also known as class variables.

Objects can also be created when a class is defined by placing their names immediately after the closing brace.

The general form is

```
        class  classname
        {
           ........;
           ........;
        } S1, S2, S3;
```

In C++ conventions of defining objects at the point of class specification is rarely followed; the user would like to define the objects as and when required, or at the point of their usage.

The following figure shows the concept:

Memory space is allocated saparately to each object for their data members. Member variables store different values for different objects of a class.

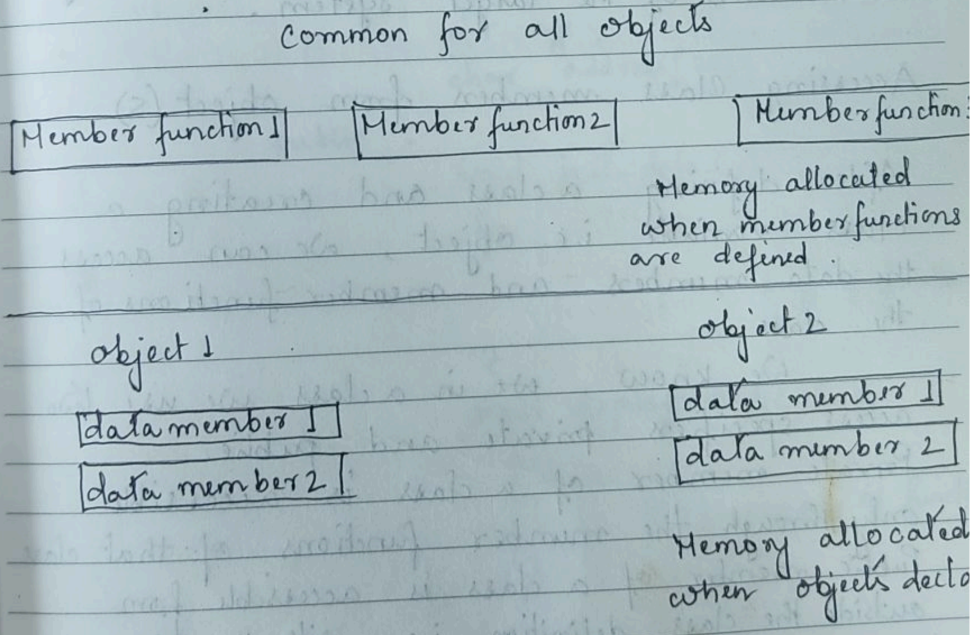

fig: A class, its member functions and objects in memory.

Date

Note:
An object is a conceptual entity possessing the following properties →

(1) It is identifiable.
(2) It has features that span a local state space.
(3) It has operations that can change the status of the system locally, while also including operations in peer objects.
(4) It refers to a thing, either tangible or a mental construct, which is identifiable by the users of the target system.

Accessing Class members from object(s).

After defining a class and creating a class variable i.e. object, we can access the data members and member functions of the class.

We know, ~~we~~ in a class we use two access specifiers private and public.
Private member of a class is accessible only through the member functions of that class
Public member of a class is accessible from outside the class definition. i.e. with the help of objects of that class and dot (.) membership operator.

Date

The general form of accessing public member is -

object_name. public_data_member_name ;

object-name. public-member-function-name (parameters);

eg

```
class student
{
        private :
                    char reg-no[10];
                    char name[30];
                    int age;
                    char address[25];

        public :
                    void init-data ( )
                    {
                        .......  // body of function
                        .......
                    }
                    void display-data()
                    {
                        .......     // body of funct
                        .......
                    }
};
```

```
void main()
{
student ob;          // class variable (object) created.
. . . . . .
. . . . . .
ob. init_data();     // Access the member function.
ob. display-data();  //      "
. . . . . .
. . . . .
}.
```

Here, the data members can be accessed in the member function as these have private scope, and the member function can be accessed outside the class i.e. before or after the main () function.

* Program to access public data member from outside the class definition.

```
#include <iostream.h>
#include <conio.h>
class test
 {
        public :
                int x,y;
 };
```

```
void main ()
{
    test t ;
    int s ;
    t.x = 10 ;
    t.y = 20 ;
    s = t.x + t.y ;
    cout << " Sum of the numbers = " << s ;
    getch();
}
```

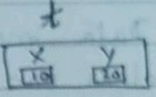

fig : memory object.

Here, test is a class with two public data members i.e. x and y. The only object t is created from class test. x and y is accessible from outside the class definition. i.e. through t, hence violating the data hiding property.

Note : If we test the program by replacing the public keyword with private. In such situation, the program will not execute, because private members are not accessible from outside the class definition.

Date

Making data member private.

Data member or member function of a class can be public, private or protected. Members of a class that is private, can be accessed only through the member function of that class.

But in general, we place all the data members in a private section and all the member function in public section.

* Program to access private data member through public member function.

```
#include<iostream.h>
#include <conio.h>
class test
  {
      int  x, y ;   // Private data member.

     public :
              void sum()                funct^n def^n inside
              {                            class def^n
                 x = 10 ;
                 y = 20 ;
                 cout << "Sum = " << x+y ;
              }
  };
```

Date

```
void main()
 {   test t;
     t. sum();
     getch();
 }
```

Here, the data members x and y becomes private, though it is not specified as private. i.e. by default members of a class becomes private.

Making member function private.

Like data member, private member function of a class can be accessed only through the public member function of that class.
Private member functions are not accessible from outside the class definition.
A member function can be called by using its name inside another member function of the same class.
This is known as nesting of member functio

Date

* Program to access private member function through public member function.

```
#include<iostream.h>
#include<conio.h>
class test
 {
        int x, y;
        void readdata()
        {     cout << "Enter the value of x";
              cin  >> x ;
              cout << "Enter the value of y";
              cin >> y ;
        }
        public :
                void disp()
                {      void readdata();
                       cout <<" x = "<< x <<"\n";
                       cout <<" y = "<< y <<"\n";
                }
 };
```

Date

```
void main ()
{
        test t;
    // t. readdata ();     // illegal, since readdata()
                              is a private member
                              functⁿ.
        t. disp();
        getch();
}
```