Ans

Refer Q 4.49 (r).

Q 4.8

How can we call member functions from outside the *class*?

Ans

To call any non-static member function, we always need to have an object of that class. To call a static member function, either an object or the class can be used.

Q 4.9

Can we access *private* data members from the outside the *class*?

Ans

No. *private* data members are only accessible from within the *class*.

There is a way in which you can access or change the *private* data of a *class* but doing so violates the object-oriented programming concepts. Following code would help you out as to how to access the private data members.

```
#include <iostream.h>

class sample
{
    private :

        int i ;

    public :

        sample ( int ii )
        {
```



```
            i = ii ;
        }

        void display( )
        {
            cout << i << endl ;
        }
};

void main( )
{
    sample s ( 97 ) ;
    s.display( ) ;
    int * p = ( int * ) &s ;
    *p = 43 ;
    s.display( ) ;
}
```

In the above example we are storing the address of the object *s* in the pointer *p*. Ideally *p* should be of the type *sample**, but we have defined it to be of type *int**. So while storing the address of object *s* we have to explicitly typecast it. After storing the address of the object *s* we can easily access the data at that address which is noting but the *private* data.

Q 4.10

How can we initialize an array of objects?

Ans

This is explained with the help of an example:

```
#include <iostream.h>

class sample
{
    int i ;
```