ASSIGNMENT 3

DATABASE

Problem Statement :-
Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving license no. etc. Construct the database with suitable member functions for initializing and destroying the data viz. constructor, default constructor, copy, constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.

Learning Objectives :-

i. Constructor
ii. Static data member and member function
iii. Inline code
iv. This pointer
v. Friend class
vi. New and delete operator
Etc.,

Theory :-

Constructor :-
A class constructor is a special member function of a class that is executed whenever we create new objects of that class.
A constructor will have exact same name as the class and it does not have any return type at all, not even void. Constructors can be very useful for setting initial values for certain member variables.

1] Default Constructor :-
If you do not define any constructor in your class then system provides it. It has no body but you can define it as well. It is also known as no argument constructor.

```
class_name()
{
        //body of the constructor
}
```

2] Parameterized Constructor :-
This type of constructor takes arguments so it is called parameterized constructor.

```
class_name(parameter list)
{
        //body of the constructor
}
```

3] Copy Constructor :-
It is a constructor with only one parameter of its same type that assigns to every non-static class member variable of the object a copy of the passed object.

```
ABC::ABC(const ABC &obj1)
{
        a = obj1.a;
        b = obj1.b;
        c = obj1.c;
}
```

4]Empty Constructor :-
It is a constructor with no parameters defined as nop (empty block of instructions).
It does nothing.
For example,

```
ABC::ABC() {}
```

Overloading Constructors :-
Like any other function, a constructor can also be overloaded with several functions that have the same name but different type or number of parameters. Remember that the compiler will execute the one that matches at the moment at which a function with that name is called.

Static member Function :-
By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::.
The same rule is applicable for static data members.
A static member function can only access static data member, other static member functions and any other functions from outside the class.
Static member functions have a class scope and they do not have access to the this pointer of the class. You could use a static member function to determine whether some objects of the class have been created or not.
Syntax :-

```
class ABC
{
        static int data;
        public:
            static void getdata();
};
```

Inline function :-
With an inline function, the compiler tries to expand the code in the body of the function in place of a call to the function.

It is useful for single line code.
For example,

```
Inline int get_x() { return x; }
```

This pointer :-
The keyword this identifies a special type of pointer. Suppose that you create an object named x of class A and class A has a non-static member function f(). If you call the function x.f(), the keyword this in the body of f() stores the address of x. You cannot declare the this pointer or make assignments to it.

```
void get_val(int num)
{
        This->num=num;
        //this pointer retrieves the value obj.num
        //this pointer is hidden by automatic variable num
}

void put_val()
{
       Cout<<num;
       //num is the value of class member
}
```

Friend Class :-
One can also declare a whole class to be a friend of another class. This allows all member functions of the friend class to have full access to all the members of other class.
Syntax:-

```
friend some_class_name;
```

Where this line is placed inside a class declaration and “some_class_name” is the name of a class known in the file holding the class declaration.
For example,

```
#include<iostream.h>

class A
{
        int data;
        public:
            A() { data = 100; }
        friend class B;
};

class B
{
        public :
```

```
        void put (A obja) { cout<<obja.data; }
};

void main()
{
        A obj1;
        B obj2;
        obj2.put(obj1);
}
```

new and delete operator :-
C++ provides the way to allocate memory dynamically (at the run time) by two memory management operator viz., new and delete.

The new operator returns a pointer to allocated memory.
Syntax :-

```
        ptr_var = new data_type(initializer);
```

The delete operator frees memory.
Syntax :-

```
        delete ptr_var;
```

Advantages :-

i. Size is computed automatically.
ii. No need for explicit type cast.
iii. new and delete can be overloaded.

Destructors :-
A mechanism, which automatically “destroys” an object when it gets invalid (for example, because of leaving scope). Destructors are used to release any resources allocated by the object`s constructor (and subsequently released in a destructor) is dynamically allocated memory. You don’t call them explicitly (they are called automatically for you), and there’s only one destructor for each object. The name of the destructor is the name of the class, preceded by a tilde(~).
For example,

```
ABC::~ABC()
{
        delete width;
        delete length;
}
```

Related Mathematics :-
//Input :-

Ai = { n, bd, bg, wt, ht }

Bi = { a, in, cn, dn }

The set A represents a set of information

Where,

N :- Name of the person

bd :- Date of birth

bd :- Blood group

wt :- Weight

ht :- Height

The set B represents a set of other information

Where,

a :- Address

in :- Insurance policy number

cn :- Contact number

dn :- Driving license number

Pi = { Ai Bi }

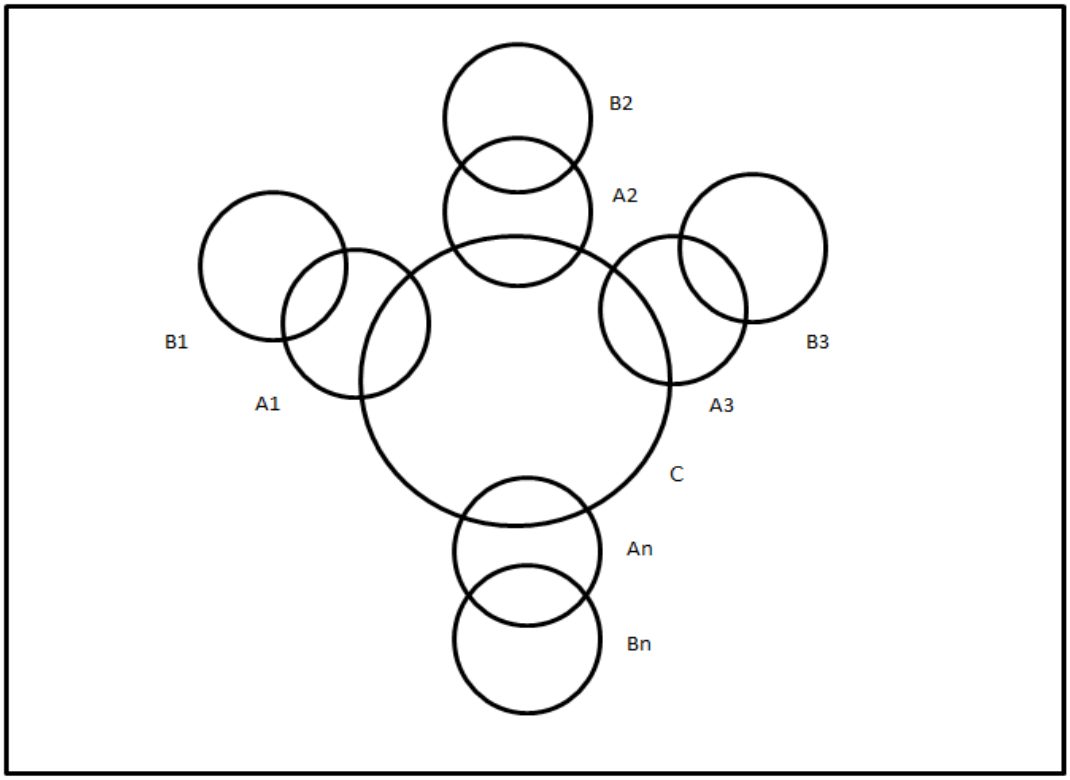
Pi represents a set which represents a set of information (Ai Bi).

It represents personnel information of the person (Pi).

//Output :-

D = { P1, P2, P3, ……, Pn }

Where the set D represents database of persons.



In this diagram, C represents the central system. In the program, the set C represents count i.e., it holds the number of persons which attached to the system.
Set A shares information with set B. The set B is attached with A. There is no connection between B and C (Central System).

Algorithm :-

1. START
2. Create class named data.
3. Declare data member as follows :
   Name, Date of birth, Blood group, Weight, Height.
4. Create count as static member of the class.
5. Create get_count() function declared as inline and static which returns count.
6. Write constructor of data class and initialize data with random values and increment count.
7. Write copy constructor of data class and transfer the values of objects.
8. Write desructor to display final message of the program.
9. Write putdata() to display data members of data class.
10. Declare extra class is friend of data class.
11. Write extra class.
12. Declare data members as follows :
    Address, Policy Number, Contact Number, Driving License Number.
13. Define constructor to initialize random values.

14. Write getdata() function to accept value of both classes from user.
15. Write putdata() function to display values of both classes.
16. Write main() function.
17. Create objects of both classes.
18. First, display demo example.
19. Call putdata() of data class. This is for detail information.
20. Call putdata() of data class by invoking copy constructor.
21. Allocate memory for object array of both classes.
22. Call getdata() function of extra class for object array.
23. Further program is based on selection logic.
24. Display menu :-
    i. Table of information.
    ii. Search
    iii. Number of records.
25. If first choice is selected, display the records in tabular form.
26. If second choice is selected, search the record and display it with details.
27. If third choice is selected, display the total count of records using get_count().
28. Frees the memory of object array of both classes.
29. STOP

Conclusion :-
By using inline function and static function, we can create counter which counts records.