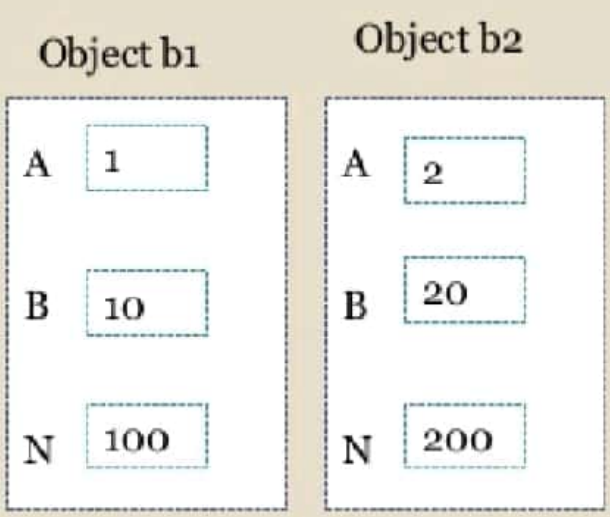
Static data member

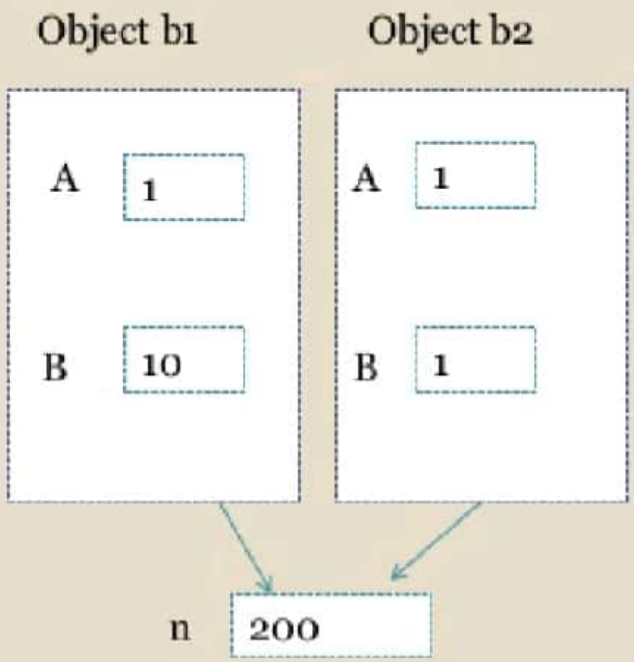
- The type of data member that is shared among all the objects of the class is known as static data members.
- Defined with static keyword
- If defined static member; only one variable is created in memory even if there are many objects of that class
- Used to share some data among all objects of a particular class
- Visible only in the class in which it is defined
- Its lifetime:
  - Starts when the program starts its execution
  - Ends when the entire program is terminated

```
class yahoo
{
    private:
            static int n;
    public:
            yahoo()
            {
                    n++;
            }
            void show()
            {
                    cout<<:you have created"<<endl<<"object so far"<<endl;
            }
};
int yahoo::n=0;
void main()
{
    yahoo x,y;
    x.show();
    yahoo z;
    x.show();
    getch();
}
```

Difference between normal and static data members



Three normal data members



Two normal data members (a,b) and one static member (n)

Static Member Function

- A function is made static by using static keyword with function name.
- It can be called using the object and the direct member access (.) operator. But, its more typical to call a static member function by itself, using class name and scope resolution (::) operator.

A function is made static by using static keyword with function name

Example:

```
class X
{
public:
static void f(){};
};
int main()
{
X::f();   // calling member function directly with class name
}
```

```
class test
{
    private:
    static int n;
    public:
    static void show()
    {
    cout<<"n = "<<n;
    }
};
int test::n = 10;
void main()
{
    test::show();
    getch();
}
```

Output

n = 10