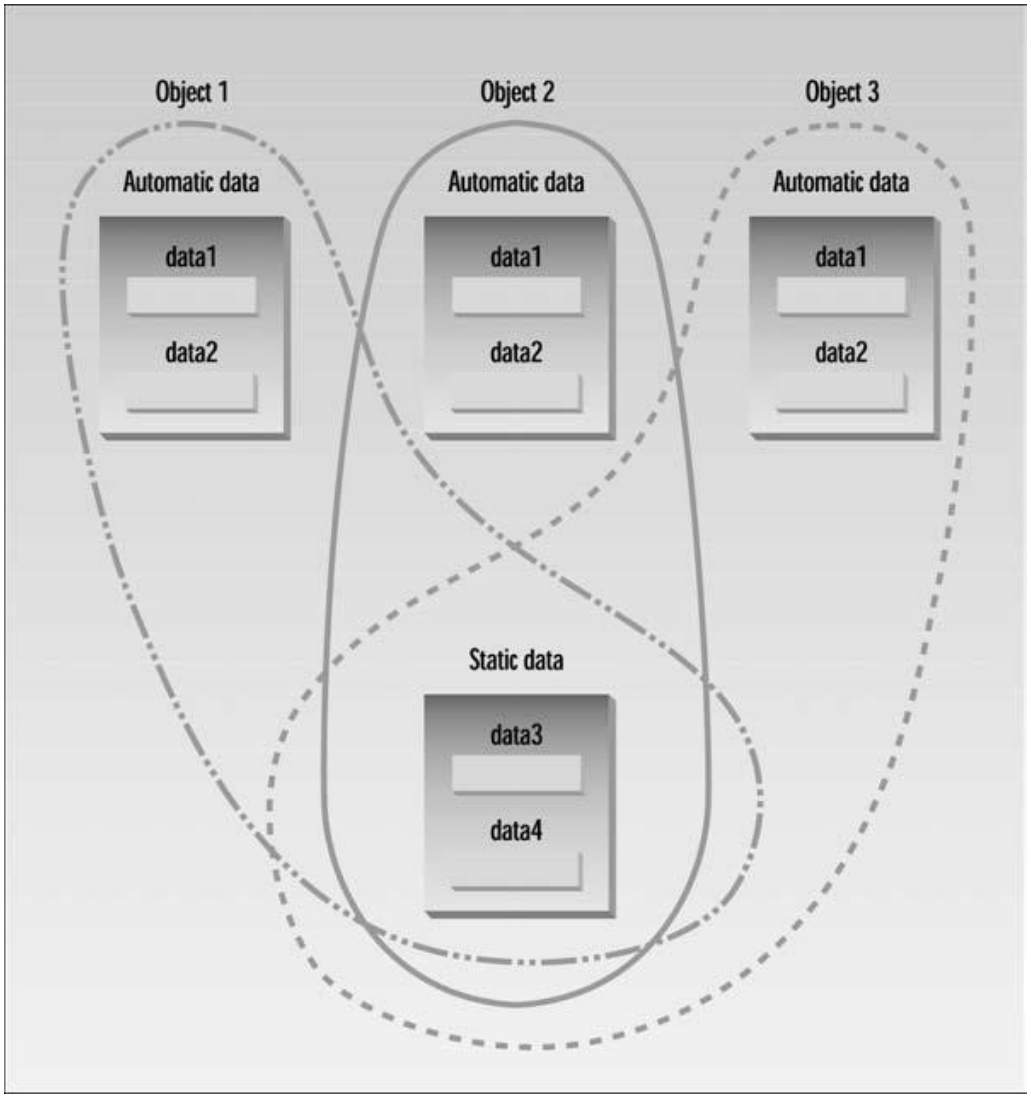
Static Class Members

- *Static fields* and *static methods* do not belong to a single instance of a class.
- A static data item is useful when all objects of the same class must share a common item of information.
- Its lifetime is the entire program. It continues to exist even if there are no objects of the class.
- To invoke a static method or a static field, use the class name, rather than the instance name.


Object 1
Automatic data
data1
data2
Object 2
Automatic data
data1
data2
Object 3
Automatic data
data1
data2
Static data
data3
data4

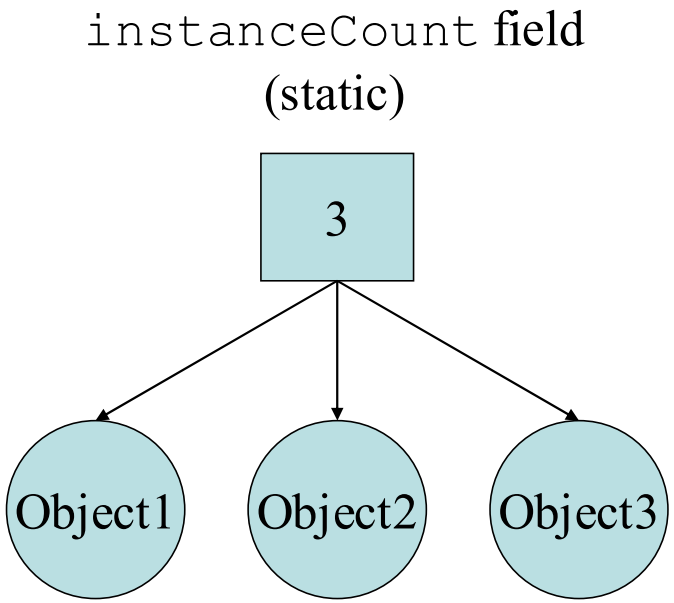
```
class Car
{

string Maker ;
int model ;
static int count;
public:
Car( ) //increments count when object created
{ count++; }
int getcount( ) //returns count
{ return count; }
};
```

```
int Car::count = 0;

int main()
{
Car c1, c2, c3; //create three objects
cout << "count is " << c1.getcount() << endl;
cout << "count is " << c2.getcount() << endl;
cout << "count is " << c3.getcount() << endl;
return 0;
}
```

Static Fields



Static Methods

- Static methods are convenient because they may be called at the class level.
- They are typically used to create utility classes.
- Static methods may not communicate with instance fields, only static fields.

```
Class Calc
{
Public:

Static int add(int num1 , int num2)
{
 return num1 + num2;
}
Static int multiply (int num1 , int num2)
{
 return num1 * num2;
}
}
```