C++ Static Members

Dr. Md. Humayun Kabir
CSE Department, BUET

Static Class Members

- Declaring static member variable as static causes only one copy of that variable to exist-no matter how many objects of that class are created
- All objects simply share that one variable
- Also, the same static variable will be shared any class derived from the class that contains the static member
- A static member exists before any of its class is created.
- A static member is a global variable that simply has its scope restricted to the class in which it is declared

Static Class Members

- A static member variable is defined with the keyword static before the data-type of the variable
- To ensure that the storage for a static member is allocated it has to be defined second time outside the class using the class-name and scope operator
- A static member is initialized to zero by default
- A static member variable can be accessed either through the objects of the class or directly through the class-name and scope resolution operator

Static Class Members

```
#include <iostream>
using namespace std;
class myclass{
    static int i;
public:
    void seti(int x){i=x;}
    int geti(){return i;}
};
int myclass::i;
//int myclass::i=100;
```

Static Class Members

```
int main(){
    myclass ob1, ob2;
    cout<<myclass::i<<endl; //Prints 0
    //cout<<myclass::i<<endl; //Prints 100
    ob1.seti(200);
    cout<<ob1.geti()<<endl; //Prints 200
    cout<<ob2.geti()<<endl; //Prints 200
    return 0;
}
```

Static Class Members

- If a member function is declared static it can access only other static members of its class
- A static member function can access global (non-static) variables and functions
- A static member function does not have a this pointer since only one copy of the function is shared by all the objects of the class
- Like static member variable, a static member function can also be accessed through the objects or through the class directly without creating any object.

Static Class Members

- Virtual static member function is not allowed
- A static member function cannot be const and volatile.

Static Class Members

```
#include <iostream>
using namespace std;
class myclass{
    static int i;
public:
    static void seti(int x){i=x;}
    int geti(){return i;}
};
int myclass::i;
```

Static Class Members

```
int main(){
    myclass ob1, ob2;
    cout<<myclass::i<<endl; //Prints 0
    ob1.seti(200);
    cout<<ob1.geti()<<endl; //Prints 200
    cout<<ob2.geti()<<endl; //Prints 200
    myclass::seti(300);
    cout<<ob1.geti()<<endl; //Prints 300
    cout<<ob2.geti()<<endl; //Prints 300
    return 0;
}
```

cout<<“Thank You”<<endl;

cout<<“**Have a Good Day**”<<endl;