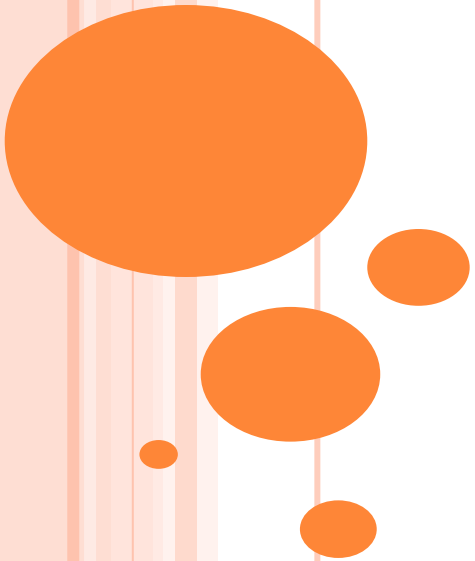


CSE 107: Object Oriented Programming Language

Dr. Tanzima Hashem
Professor
CSE, BUET

Static Class Members

- Static member variable
  - 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 by any class derived from the class that contains the static member
  - Exists before any of its object is created
  - A global variable that simply has its scope restricted to the class in which it is declare

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 variable is initialized to zero by default
- A static member variable can be accessed
  - 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 main(){
    myclass ob1, ob2;
    ob1.seti(200);
    cout<<ob1.geti()<<endl;
    //Prints 200
    cout<<ob2.geti()<<endl;
    //Prints 200
    return 0;
}
```

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=500; // allowed
```

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

Static Class Members

```
#include <iostream>
using namespace std;
class myclass{

public:
        static int i;
        void seti(int x){i=x;}
        int geti(){return i;}
};

int myclass::i=100;
```

```
int main(){
    myclass ob1, ob2;
    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

- Static member function
  - Access only other static members of its class
  - 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

- A static member function can be accessed
  - through the objects or
  - through the class directly without creating any object

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

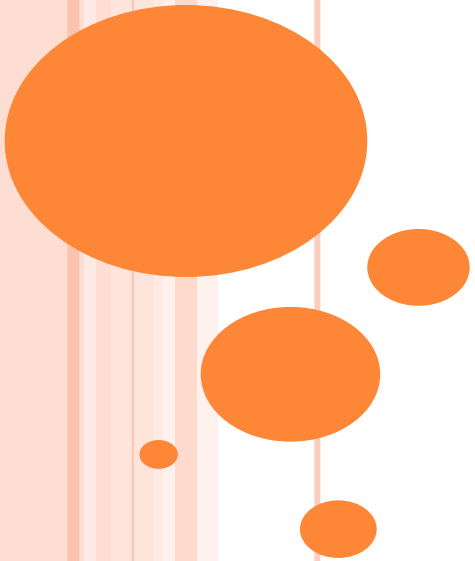
Static Class Members

```
#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;
```

```
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;
}
```

Acknowledgement
http://faizulbari.buet.ac.bd/Courses.html
http://mhkabir.buet.ac.bd/cse201/index.html



The End

Topic Covered: Section 13.3