

---

## Static Data Members, and Static Member Functions

---

---

---

---

---

---

---

---

### **static** Class Members

---

- Each object gets it's own copy of the data members
- What if we wanted a data member to be shared between all objects
  - Each object sees the same value for the data member
  - Each object can modify that data member, and the other objects will see the change
- Data members of this type are called static

---

---

---

---

---

---

---

---

### **static** Class Member (11.2)

---

- **static** members represent class-wide information and are not specific to one object
- There is only one copy of the member and it is shared between all objects
- Why would we ever need or want a static class member? Can you think of an example.

---

---

---

---

---

---

---

---

## static Class Members

- They are not global variables
- The static data member could be declared public, private, or protected
- static data members must be initialized once

---

---

---

---

---

---

---

---

## Example

```
#ifndef EMPLOYEE_H
#define EMPLOYEE_H
class Employee
{
private:
    char *firstName;
    char *lastName;
    static int count;
public:
    Employee (const char *,const char *);
    ~Employee ();
    char *getFirstName () const;
    char *getLastName () const;
    static int getCount ();
};
#endif
```

---

---

---

---

---

---

---

---

## Constructor Definition

```
Employee::Employee (const char * first,
                    const char * last)
{
    firstName = new char[strlen(first) + 1];
    strcpy (firstName, first);
    lastName = new char[strlen(last) + 1];
    strcpy (lastName, last);
    count++;
}
```

---

---

---

---

---

---

---

---

What is the value of count?

```
int Employee::count = 0;
int main()
{
    Employee emp1 ("john", "doe");
    Employee emp2 ("jane", "doe");
    Employee emp3 ("bob", "doe");
}
```

---

---

---

---

---

---

---

---

static Member Functions

```
class IntVal
{
    private:
        int value;
        static int valCount;
    public:
        static int getValCount()
        { return valCount; }
};
```

---

---

---

---

---

---

---

---

Calling Static Functions

- Can be called independently of class objects, through the class name:
- ```
cout << IntVal::getValCount();
```
- Can be called before any objects of the class have been created
  - Used mostly to manipulate static member variables of the class

---

---

---

---

---

---

---

---