Friend Functions

Sections 11.3, 11.4, 11.5

`friend` Functions

- Only the member functions of a class have direct access to the private data members of the class
- **`friend`** functions are friends of the class that are defined outside of the class but still have access to private data members

`friend` Functions

- The function prototype is placed in the class, preceded by the keyword **`friend`**
- The function definition can be written anywhere without the class name (class::)
- The function is able to directly access the private data members

`friend` Functions

- The `friend` function could be a member function in another class
- A class could also be made a friend of an existing class
  - In this case, every member function of the friend class will have access to this class's private data

Example

- Let us investigate program 11-4 on pages 670-674
- This program is used to keep track of the budgets of divisions and auxiliary departments of these divisions

Recall from Lecture 8

```
Time cTest1(9, 25, 32);
Time cTest2;

cTest2 = cTest1;

cTest2.printStandard();
```

Copy Constructors

- Default copy constructors are included in all classes
- They allow a new object to be created that is a copy of an existing object
- Example:
  - `Time cTeaTime(16, 0, 0);`
  - `Time cGameTime = cTeaTime;`
- What deficiencies could this have?

Copy Constructors

- What if the class contained a pointer data member?
- Look at program 11-7 on pages 679-681
- How could the program be modified?
  - Add a copy constructor

Constructor Prototype

- First, uncomment the destructor
  - Why was it commented in the original program?
- Add the following prototype to NumberArray.h

```
NumberArray(NumberArray &);
```

Copy Constructor Definition

```
NumberArray::NumberArray(NumberArray &obj)
{
  arraySize = obj.arraySize;
  aPtr = new double[arraySize];
  for(int index = 0; index < arraySize; index++)
  {
    aPtr[index] = obj.aPtr[index];
  }
}
```