Inheritance

What is it?

- Inheritance can be thought of as software reusability where one class inherits another classes' data and methods and adds new functionality of its own
- Parts:
  - superclass - the existing class
  - subclass - the new class with inherited members and additional behaviors

Public Inheritance

- Every derived class object is also an object of the superclass.
- As an example, if the superclass is "Vehicle" then a subclass might be "Cars" and "Trucks." Cars inherit the members and behaviors of a Vehicle and add other behaviors and members
- Members of a subclass cannot directly access the private members of a superclass

Example

```
Class Person
{
  private:
    string name;
  public:
    Person() { setName(""); }
    Person(string pName) { setName(pName); }
    void setName(string pName) { name = pName;
  }
    string getName() { return name; }
};
```


Enumerated Data Types (4.14)

- Enumerated data types are programmer-defined data type that contain a set of named integer constants

```
enum Roster{ Abbey, Brittany,
  Stephanie, Brandon, Nick,
  Jacquie, Chris, Erik, Michaela,
  Reid, Josh, Perry, Monica, Maria,
  Max };

Roster student;

student = Chris;
```


Enumerations

- We are to create two enumerations to be used with the Person class as follows:

```
enum Discipline { MATH, BIOLOGY,
  COMPUTER_SCIENCE };

enum Classification { FRESHMAN,
  SOPHOMORE, JUNIOR, SENIOR };
```

Another Class

```
class Student : public Person
{
  private:
    Discipline major;
    Person *advisor;
  public:
    void setMajor(Discipline d) { major = d; }
    Discipline getMajor() { return major; }
    void setAdvisor(Person *p) { advisor = p; }
    Person *getAdvisor() { return advisor; }
};
```


Yet Another Class

```
class Faculty : public Person
{
  private:
    Discipline department;
  public:
    void setDepartment(Discipline d) {
  department = d; }
    Discipline getDepartment() { return
  department; }
};
```


So, how can they be used?

```
const string dName[] = { "Math", "Bio", "CS"};
const string cName[] = { "Freshman", "Sophomore",
  "Junior", "Senior" };
int main()
{
  Faculty prof;
  prof.setName("Indianan Jones");
  prof.setDepartment(MATH);
  cout << "Prof." << prof.getName() << " teaches in ";
  Discipline dept = prof.getDepartment();
  cout << dName[dept] << endl;
  return 0;
}
```