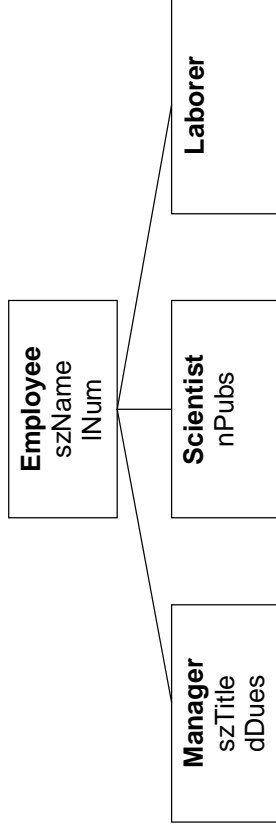


# CST8130: Data Structures: Inheritance

## Program 1: Inheritance Model



// employ.cpp -- models employee database using inheritance  
#include <iostream>  
const int LEN = 80;

```

class Employee {
private:
char szName[LEN];
unsigned long INum;
public:
void GetData()
{
cout << "\n Enter last szName: "; cin >> szName;
cout << " Enter INum: "; cin >> INum;
}
void PutData() const
{
cout << "\n Name: " << szName;
cout << "\n Number: " << INum;
}
};
  
```

```

class Manager : public Employee {
private:
char szTitle[LEN]; // "vice-president" etc.
double dDues; // golf club dues
public:
void GetData()
{
Employee::GetData();
cout << " Enter szTitle: "; cin >> szTitle;
cout << " Enter golf club dues: "; cin >> dDues;
}
void PutData() const
{
Employee::PutData();
cout << "\n Title: " << szTitle;
cout << "\n Golf club dues: " << dDues;
}
};
  
```

```

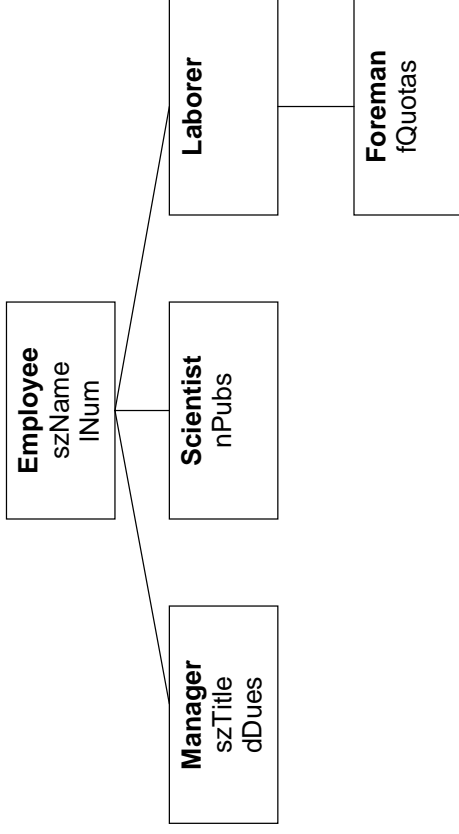
class Scientist : public Employee {
private:
int nPubs; // Number of publications
public:
void GetData()
{
Employee::GetData();
cout << " Enter number of pubs: "; cin >> nPubs;
}
void PutData() const
{
Employee::PutData();
cout << "\n Number of publications: " << nPubs;
}
};
  
```

```
class Laborer : public Employee {  
};
```

```
void main()  
{  
    Manager m1, m2;  
    Scientist s1;  
    Laborer l1;  
  
    cout << endl;  
    cout << "\nEnter data for Manager 1"; m1.GetData();  
    cout << "\nEnter data for Manager 2"; m2.GetData();  
    cout << "\nEnter data for Scientist 1"; s1.GetData();  
    cout << "\nEnter data for Laborer 1"; l1.GetData();  
  
    cout << "\nData on Manager 1"; m1.PutData();  
    cout << "\nData on Manager 2"; m2.PutData();  
    cout << "\nData on Scientist 1"; s1.PutData();  
    cout << "\nData on Laborer 1"; l1.PutData();  
    cout << endl;  
}
```

### **Program 1: Memory Map**

### Program 2: Inheritance Model



```

void main()
{
  Laborer l1;
  Foreman f1;

  cout << endl;
  cout << "\nEnter data for Laborer 1"; l1.GetData();
  cout << "\nEnter data for Foreman 1"; f1.GetData();

  cout << endl;
  cout << "\nData on Laborer 1"; l1.PutData();
  cout << "\nData on Foreman 1"; f1.PutData();

  cout << endl;
}
  
```

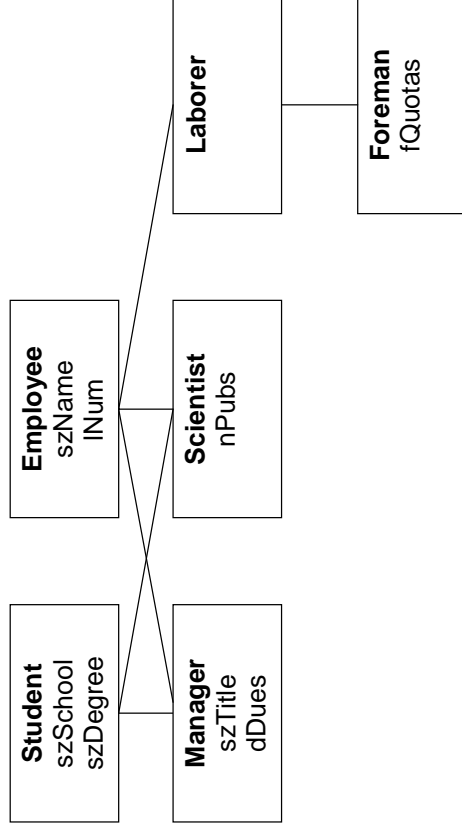
### Program 2: Memory Map

```

class Foreman : public Laborer {
private:
  float fQuotas;
public:
  void GetData()
  {
    Laborer::GetData();
    cout << " Enter quotas: "; cin >> fQuotas;
  }
  void PutData() const
  {
    Laborer::PutData();
    cout << "\n Quotas: " << fQuotas;
  }
};
  
```

### Additions / Modifications to Program 1

### Program 4: Multiple Inheritance Model



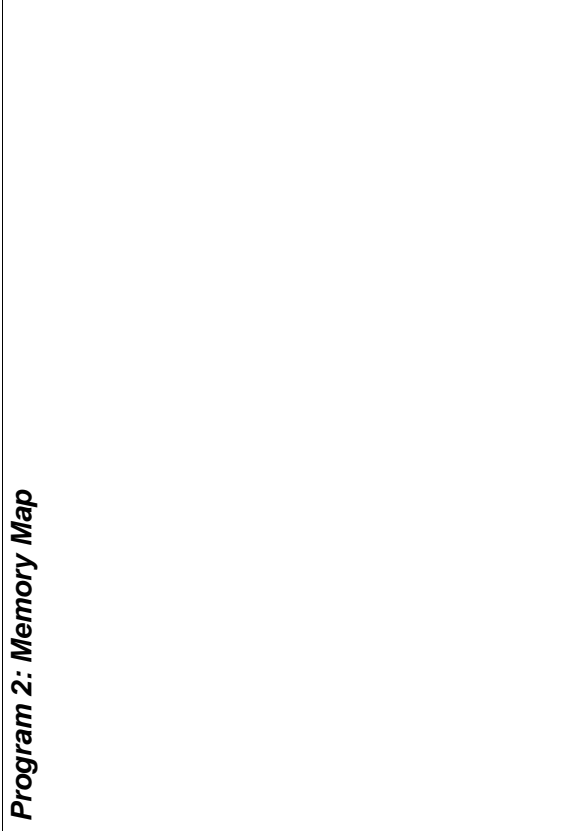
```

// empmult.cpp -- multiple inheritance employees and degrees
#include <iostream>
const int LEN = 80;
  
```

```

class Student { // educational background
private:
    char szSchool[LEN];
    char szDegree[LEN];
public:
    void GetEdu()
    {
        cout << "Enter name of college: ";   cin >> szSchool;
        cout << "Enter highest degree: ";   cin >> szDegree;
    }
    void PutEdu() const
    {
        cout << "\nCollege: " << szSchool << "\nDegree: " << szDegree;
    }
};
  
```

### Program 2: Memory Map



```

class Employee {
private:
    char szName[LEN];
    unsigned long INumber;
public:
    void GetData()
    {
        cout << "\nEnter last name: ";   cin >> szName;
        cout << "Enter number: ";   cin >> INumber;
    }
    void PutData() const
    {
        cout << "\nName: " << szName << "\nNumber: " << INumber;
    }
};
  
```

```

class Manager : private Employee, private Student {
private:
char szTitle[LEN];
double dDues;
public:
void GetData()
{
Employee::GetData();
cout << "Enter title: "; cin >> szTitle;
cout << "Enter golf club dues: "; cin >> dDues;
Student::GetEdu();
}
void PutData() const
{
Employee::PutData();
cout << "nTitle: " << szTitle << "nGolf club dues: " << dDues;
Student::PutEdu();
}
};

```

```

class Laborer : public Employee {
};

void main()
{
Manager m1;
Scientist s1, s2;
Laborer l1;

cout << "nEnter data for manager 1"; m1.GetData();
cout << "nEnter data for scientist 1"; s1.GetData();
cout << "nEnter data for scientist 2"; s2.GetData();
cout << "nEnter data for laborer 1"; l1.GetData();

cout << "nData on manager 1"; m1.PutData();
cout << "nData on scientist 1"; s1.PutData();
cout << "nData on scientist 2"; s2.PutData();
cout << "nData on laborer 1"; l1.PutData();
}

```

```

class Scientist : private Employee, private Student {
private:
int nPubs;
public:
void GetData()
{
Employee::GetData();
cout << "Enter number of pubs: "; cin >> nPubs;
Student::GetEdu();
}
void PutData() const
{
Employee::PutData();
cout << "nNumber of publications: " << nPubs;
Student::PutEdu();
}
};

```