```
#include <string>
using namespace std;

enum Discipline { ARCHEOLOGY, BIOLOGY, COMPUTER_SCIENCE };
enum Classification { FRESHMAN, SOPHOMORE, JUNIOR, SENIOR };


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


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


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

```
// This program demonstrates the creation and use
// of objects of derived classes.

#include <iostream>
#include "inheritance.h"

using namespace std;


// These arrays of string are used to print the
// enumerated types.
const string dName[] = {
     "Archeology", "Biology", "Computer Science"
};

const string cName[] = {
     "Freshman", "Sophomore", "Junior", "Senior"
};



int main() {
   // Create a Faculty object
   Faculty prof;

   // Use a Person member function
   prof.setName("Indiana Jones");

   // Use a Faculty member function
   prof.setDepartment(ARCHEOLOGY);
   cout << "\nProfessor " << prof.getName()
        << " teaches in the " << "Department of ";

   // Get department as an enumerated type
   Discipline dept = prof.getDepartment();

   // Print out the department in string form
   cout << dName[dept] << endl << endl;

   return 0;
}
```

OUTPUT:

```
Professor Indiana Jones teaches in the Department of Archeology
```