Inheritance and Function Overriding
(Winter 2015)

Write a C++ program to define an `Employee` class, a `FacultyMember` class as a derived class of the `Employee` class, and a test driver program. The `Employee` class stores the employee's `firstName`, `lastName`, and `SIN` (Social Insurance Number). This information is common to all employees including those in `FacultyMember` derived class. The `Employee` class should provide three interfaces--- a constructor, a destructor, and a `print`. The constructor function receives the `firstName`, `lastName`, and `SIN` strings to be stored. The only access to the data must be through the `print` interface which simply outputs the first name, last name, and SIN of the employee.

Class `FacultyMember` inherits from `Employee` class with public inheritance. The `FacultyMember` class includes its own interfaces `print` and `getCourses`. The `print` function must have the same prototype as the `Employee print` --- this is an example of function overriding. Therefore, class `FacultyMember` has access to two `print` functions. Class `FacultyMember` also contains data members `office` and `courses` for storing the employee's office number and courses taught during the year. For simplicity of programming, in this exercise, we assume that all faculty members are full-time tenured, and the courses are identified by their subject and number (e.g. ENGI 3050). The `FacultyMember` constructor must pass the first name, last name, and SIN strings to the `Employee` constructor so the base class data members `firstName`, `lastName` and `SIN` can be initialized. It then initializes its own data members `office` and `courses`. The member functions `setOffice` and `getOffice`, and `setCourses` and `getCourses` are declared in class `FacultyMember` to set and access the faculty member's office number and courses taught. The `FacultyMember print` function must call the `Employee print` function to output the employee's first and last names and the SIN. The `FacultyMember print` function must also output the employee's office number and courses taught during the year.

Your test driver program must instantiate some `FacultyMember` objects, initialize them, and for each object, call the `FacultyMember print` function to output the object's data.