4B3G0123 資工一丙 黃正豪

The result:

```
Student Details:
ID: S001
Name: Anna Smith
Gender: Female
Birth Date: 2002-05-14
-------------------
Teacher Details:
ID: T001
Name: James Brown
Gender: Male
Birth Date: 1980-09-21
Courses:
Course ID: C101, Course Name: Mathematics
Course ID: C102, Course Name: Physics
```

Dec04.cpp

```cpp
1   #include <iostream>
2   #include <string>
3   #include <vector>
4
5   // Base class: Person
6   class Person {
7   public:
8       std::string ID;
9       std::string LastName;
10      std::string FirstName;
11      std::string Gender;
12      std::string BirthDate;
13
14      // Constructor
15      Person(const std::string& id, const std::string& lastName, const std::string& firstName,
16              const std::string& gender, const std::string& birthDate)
17          : ID(id), LastName(lastName), FirstName(firstName), Gender(gender),
    BirthDate(birthDate) {}
18
19      virtual ~Person() {} // Virtual destructor for polymorphic behavior
20  };
21
22  // Class: Student
23  class Student : public Person {
24  public:
25      // Constructor
26      Student(const std::string& id, const std::string& lastName, const std::string& firstName,
27              const std::string& gender, const std::string& birthDate)
28          : Person(id, lastName, firstName, gender, birthDate) {}
29
30      void Output() {
31          std::cout << "Student Details:\n"
32                    << "ID: " << ID << "\n"
33                    << "Name: " << FirstName << " " << LastName << "\n"
34                    << "Gender: " << Gender << "\n"
35                    << "Birth Date: " << BirthDate << "\n";
36      }
37  };
38
39  // Class: Course
40  class Course {
41  public:
42      std::string CourseID;
43      std::string CourseName;
44
45      // Constructor
46      Course(const std::string& courseId, const std::string& courseName)
47          : CourseID(courseId), CourseName(courseName) {}
```

```cpp
48
49        void DisplayCourse() {
50            std::cout << "Course ID: " << CourseID << ", Course Name: " << CourseName << "\n";
51        }
52    };
53
54    // Class: Teacher
55    class Teacher : public Person {
56    private:
57        std::vector<Course> courses; // Vector to store courses taught by the teacher
58
59    public:
60        // Constructor
61        Teacher(const std::string& id, const std::string& lastName, const std::string& firstName,
62                const std::string& gender, const std::string& birthDate)
63            : Person(id, lastName, firstName, gender, birthDate) {}
64
65        // Add a course
66        void AddCourse(const Course& course) {
67            courses.push_back(course);
68        }
69
70        // Output teacher details and courses
71        void Output() {
72            std::cout << "Teacher Details:\n"
73                      << "ID: " << ID << "\n"
74                      << "Name: " << FirstName << " " << LastName << "\n"
75                      << "Gender: " << Gender << "\n"
76                      << "Birth Date: " << BirthDate << "\n"
77                      << "Courses:\n";
78
79            for (const auto& course : courses) {
80                course.DisplayCourse();
81            }
82        }
83    };
84
85    int main() {
86        // Create a Student object
87        Student student1("S001", "Smith", "Anna", "Female", "2002-05-14");
88        student1.Output();
89
90        std::cout << "------------------\n";
91
92        // Create a Teacher object
93        Teacher teacher1("T001", "Brown", "James", "Male", "1980-09-21");
94
95        // Add courses to the teacher
96        teacher1.AddCourse(Course("C101", "Mathematics"));
97        teacher1.AddCourse(Course("C102", "Physics"));
```

```cpp
 98
 99        teacher1.Output();
100
101        return 0;
102    }
103
```