Dec04.cpp

```cpp
#include <iostream>
#include <string>
#include <vector>

// Base class: Person
class Person {
public:
    std::string ID;
    std::string LastName;
    std::string FirstName;
    std::string Gender;
    std::string BirthDate;

    // Constructor
    Person(const std::string& id, const std::string& lastName, const std::string& firstName,
           const std::string& gender, const std::string& birthDate)
        : ID(id), LastName(lastName), FirstName(firstName), Gender(gender), BirthDate(birthDate) {}

    virtual ~Person() {} // Virtual destructor for polymorphic behavior
};

// Class: Student
class Student : public Person {
public:
    // Constructor
    Student(const std::string& id, const std::string& lastName, const std::string& firstName,
            const std::string& gender, const std::string& birthDate)
        : Person(id, lastName, firstName, gender, birthDate) {}

    void Output() {
        std::cout << "Student Details:\n"
                  << "ID: " << ID << "\n"
                  << "Name: " << FirstName << " " << LastName << "\n"
                  << "Gender: " << Gender << "\n"
                  << "Birth Date: " << BirthDate << "\n";
    }
};

// Class: Course
class Course {
public:
    std::string CourseID;
    std::string CourseName;

    // Constructor
    Course(const std::string& courseId, const std::string& courseName)
        : CourseID(courseId), CourseName(courseName) {}

    void DisplayCourse() {
        std::cout << "Course ID: " << CourseID << ", Course Name: " << CourseName << "\n";
    }
};

// Class: Teacher
class Teacher : public Person {
private:
    std::vector<Course> courses; // Vector to store courses taught by the teacher

public:
    // Constructor
    Teacher(const std::string& id, const std::string& lastName, const std::string& firstName,
            const std::string& gender, const std::string& birthDate)
        : Person(id, lastName, firstName, gender, birthDate) {}

    // Add a course
    void AddCourse(const Course& course) {
        courses.push_back(course);
    }

    // Output teacher details and courses
    void Output() {
        std::cout << "Teacher Details:\n"
                  << "ID: " << ID << "\n"
                  << "Name: " << FirstName << " " << LastName << "\n"
                  << "Gender: " << Gender << "\n"
                  << "Birth Date: " << BirthDate << "\n"
                  << "Courses:\n";

        for (const auto& course : courses) {
            course.DisplayCourse();
        }
    }
};

int main() {
    // Create a Student object
    Student student1("S001", "Smith", "Anna", "Female", "2002-05-14");
    student1.Output();

    std::cout << "-------------------\n";

    // Create a Teacher object
    Teacher teacher1("T001", "Brown", "James", "Male", "1980-09-21");

    // Add courses to the teacher
    teacher1.AddCourse(Course("C101", "Mathematics"));
    teacher1.AddCourse(Course("C102", "Physics"));

    teacher1.Output();

    return 0;
}

```