

Dec04.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),
18           BirthDate(birthDate) {}
19
20     virtual ~Person() {} // Virtual destructor for polymorphic behavior
21 };
22
23 // Class: Student
24 class Student : public Person {
25 public:
26     // Constructor
27     Student(const std::string& id, const std::string& lastName, const std::string& firstName,
28           const std::string& gender, const std::string& birthDate)
29         : Person(id, lastName, firstName, gender, birthDate) {}
30
31     void Output() {
32         std::cout << "Student Details:\n"
33             << "ID: " << ID << "\n"
34             << "Name: " << FirstName << " " << LastName << "\n"
35             << "Gender: " << Gender << "\n"
36             << "Birth Date: " << BirthDate << "\n";
37     }
38 };
39
40 // Class: Course
41 class Course {
42 public:
43     std::string CourseID;
44     std::string CourseName;
45
46     // Constructor
47     Course(const std::string& courseId, const std::string& courseName)
48         : CourseID(courseId), CourseName(courseName) {}
```

```
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"));
```

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