

# Inheritance

## Lab Assignment: Course information (derived classes)

Given main(), define a **Course** base class with functions to set and get the private data members of the following types:

- string to store the course number
- string to store the course title

Define **Course**'s PrintInfo() function that outputs the course number and title.

Then, define a derived class **OfferedCourse** with functions to set and get the private data members of the following types:

- string to store the instructor name
- string to store the location
- string class time

Ex. If the input is:

```
In [ ]: ECE287
Digital Systems Design
ECE387
Embedded Systems Design
Mark Patterson
Wilson Hall 231
WF: 2-3:30 pm
```

The output is

```
In [ ]: Course Information:
Course Number: ECE287
Course Title: Digital Systems Design
Course Information:
Course Number: ECE387
Course Title: Embedded Systems Design
Instructor Name: Mark Patterson
Location: Wilson Hall 231
Class Time: WF: 2-3:30 pm
```

File is marked as read only

Current file: **main.cpp**

```
1 #include "OfferedCourse.h"
2
3 int main() {
4     Course myCourse;
5     OfferedCourse myOfferedCourse;
6
7     string courseNumber, courseTitle;
8     string oCourseNumber, oCourseTitle, instructorName, location, classTime;
9
10    getline(cin, courseNumber);
11    getline(cin, courseTitle);
12
13    getline(cin, oCourseNumber);
14    getline(cin, oCourseTitle);
15    getline(cin, instructorName);
16    getline(cin, location);
17    getline(cin, classTime);
18
19    myCourse.SetCourseNumber(courseNumber);
20    myCourse.SetCourseTitle(courseTitle);
21    myCourse.PrintInfo();
22
23    myOfferedCourse.SetCourseNumber(oCourseNumber);
24    myOfferedCourse.SetCourseTitle(oCourseTitle);
25    myOfferedCourse.SetInstructorName(instructorName);
26    myOfferedCourse.SetLocation(location);
27    myOfferedCourse.SetClassTime(classTime);
28    myOfferedCourse.PrintInfo();
29
30    cout << "    Instructor Name: " << myOfferedCourse.GetInstructorName() << endl;
31    cout << "    Location: " << myOfferedCourse.GetLocation() << endl;
32    cout << "    Class Time: " << myOfferedCourse.GetClassTime() << endl;
33 }
34
```

Current file: **Course.h**

```
1 #ifndef COURSEH
2 #define COURSEH
3
4 #include <iostream>
5 #include <string>
6
7 using namespace std;
8
9 class Course {
10    // TODO: Declare private data members
11
12
13    // TODO: Declare mutator functions
14    //      SetCourseNumber(), SetCourseTitle()
15
16
17    // TODO: Declare accessor functions -
18    //      GetCourseNumber(), GetCourseTitle()
19
20
21    // TODO: Declare PrintInfo()
22
23 };
24
25 #endif
26
```

Current file: **OfferedCourse.h**

```
1 #ifndef OFFERED_COURSEH
2 #define OFFERED_COURSEH
3
4 #include "Course.h"
5
6 class OfferedCourse : public Course {
7    // TODO: Declare private data members
8
9
10    // TODO: Declare mutator functions -
11    //      SetInstructorName(), SetLocation(), SetClassTime()
12
13
14    // TODO: Declare accessor functions -
15    //      GetInstructorName(), GetLocation(), GetClassTime()
16
17
18 };
19
20 #endif
```

Current file: **Course.cpp**

```
1 #include "Course.h"
2
3 // TODO: Define mutator functions -
4 //      SetCourseNumber(), SetCourseTitle()
5
6
7 // TODO: Define accessor functions -
8 //      GetCourseNumber(), GetCourseTitle()
9
10
11 // TODO: Define PrintInfo()
```

Current file: **OfferedCourse.cpp**

```
1 #include "OfferedCourse.h"
2
3 // TODO: Define mutator functions -
4 //      SetInstructorName(), SetLocation(), SetClassTime()
5
6
7 // TODO: Define accessor functions -
8 //      GetInstructorName(), GetLocation(), GetClassTime()
```

## Assignment Tests:

Apply the following 5 tests for 10 points

### 1. Compare output (1 point)

When input is

```
ECE287
Digital Systems Design
ECE387
Embedded Systems Design
Mark Patterson
Wilson Hall 231
WF: 2-3:30 pm
```

Standard output exactly matches

```
Course Information:
Course Number: ECE287
Course Title: Digital Systems Design
Course Information:
Course Number: ECE387
Course Title: Embedded Systems Design
Instructor Name: Mark Patterson
Location: Wilson Hall 231
Class Time: WF: 2-3:30 pm
```

### 2. Compare output (1 point)

When input is

```
CSE 174
Systems I
CSE 274
Systems II
Dr. Susan Thomas
MSE 108
MWF: 10-10:50 am
```

Standard output exactly matches

```
Course Information:
Course Number: CSE 174
Course Title: Systems I
Course Information:
Course Number: CSE 274
Course Title: Systems II
Instructor Name: Dr. Susan Thomas
Location: MSE 108
Class Time: MWF: 10-10:50 am
```

### 3. Compare output (1 point)

When input is

```
CEC 101
Introduction to Computing
CEC 102
Computing and beyond
Dr. Rob Adams
Pierce Hall 56
MWF: 3-4:50 pm
```

Standard output exactly matches

```
Course Information:
Course Number: CEC 101
Course Title: Introduction to Computing
Course Information:
Course Number: CEC 102
Course Title: Computing and beyond
Instructor Name: Dr. Rob Adams
Location: Pierce Hall 56
Class Time: MWF: 3-4:50 pm
```

### 4. Compare output (1 point)

When input is

```
ECE201
Circuits I
ECE301
Circuits II
Jeff Peters
Univ. Center 147
WF: 12-1:30 pm
```

Standard output exactly matches

```
Course Information:
Course Number: ECE201
Course Title: Circuits I
Course Information:
Course Number: ECE301
Course Title: Circuits II
Instructor Name: Jeff Peters
Location: Univ. Center 147
Class Time: WF: 12-1:30 pm
```

### 5. Compare output (1 point)

When input is

```
CSE101
Algorithm I
CSE102
Algorithm II
Tim Allen
Sondheim Hall 333
WF: 1-2:30 pm
```

Standard output exactly matches

```
Course Information:
Course Number: CSE101
Course Title: Algorithm I
Course Information:
Course Number: CSE102
Course Title: Algorithm II
Instructor Name: Tim Allen
Location: Sondheim Hall 333
Class Time: WF: 1-2:30 pm
```

## Submissions

Note: Do not forget to submit the assignment and its corresponding test outputs to receive full credit.

1 - Name your C++ files FirstName\_Lastname\_Course\_Information.cpp.

2 - Prepare your report in docx or pdf format and name it Firstname\_Lastname.docx or Firstname\_Lastname.pdf.

3 - Add the screenshot of your code to the report and provide a description for it. All tests should be performed and the result screenshot be included in the report.

Note: Make sure to have your report containing both explanatnations and screenshots.