

# Homework 3: Classes and Lists

Due: Tuesday, Feb 11, 11:55pm

How to turn in:

- Write two programs and submit them on iLearn
- If you use repl.it, the files will be named main.cpp; you will need to rename them
- The names are in the assignment; incorrect names will be graded as 0

## Part 1. Course Roster (15 points)

Write a program to define and use two classes: Course and Student.

A Student should have the following member variables:

- string name
- int id

And the following member functions:

- Default constructor (THIS IS IMPLIED; ALWAYS CREATE ONE, EVEN IF UNSTATED)
- A constructor that takes a name and an id
- Get methods for name and id
- Set method for name, but NOT id

The Course class should have the following member variables:

- string title
- int number
- string instructorName
- an array of student objects (up to 30)
- The number of enrolled students

And the following member functions:

- Do you need a default constructor?
- A constructor that takes a course title, a course number, and a course instructor name
- Member functions to get the title, instructor name, and number
- `bool add(Student s);`

A member function to add a Student. It should verify that the student has not already been added by checking if the id is a repeat; if the student is already in the course or the course is full, return false

- `bool remove(Student s);`

A member function to remove a Student. If the student is in the course, it should remove them, shift the remaining students to the left as necessary, lower the number number of enrolled students, and return true. If the student is not enrolled, return false

- `void printAll() const;`

A function to print out all of the course information, including a list of students.

Here is a sample test program:

```
int main() {
    Course c1("Intro to Everything", 123, "Dr. Gross");
    Student s1("Jane", 1234);
    Student s2("Phillipe", 1235);
    Student s3("Lu", 1236);
    Student s4("Enrique", 1237);
    c1.add(s1);
    c1.add(s2);
    c1.add(s3);
    c1.add(s4);
    c1.printAll();
    c1.remove(s2);
    c1.printAll();
}
```

A sample run of this program should look like this:

```
Course Title: Intro to Everything
Course Number: 123
Instructor: Dr. Gross
Enrollment: 4
    Jane: 1234
    Phillipe: 1235
    Lu: 1236
    Enrique: 1237
```

```
Course Title: Intro to Everything
Course Number: 123
Instructor: Dr. Gross
Enrollment: 3
    Jane: 1234
    Lu: 1236
    Enrique: 1237
```

Final Notes:

- Read the sample run carefully to understand the requirements of the problem correctly.
- This program should be uploaded to iLearn as hw03-1.cpp

## Part 2. Even More New List Operations (15 points)

Starting with your int List, you will add the following operations:

The function `append()` adds the "item" at the end of the list.

- `int getSize() const;`  
This function returns the size of the list (`mySize`).

- `int getValue(int pos) const;`  
This function returns the value of the element at the specified position. If the position is unused, it returns -1.
- `bool hasDuplicates() const;`  
This function checks if there are any duplications in the list, returning true if there are. For example, if a list contains 10, 30, and 20, the function should return false. However, if the list has 10, 30, 20, and 30, it will return true.
- `bool isSorted() const;`  
This function checks if the list is sorted in ascending order. For example, if a list contains 10, 30, and 20, the function should return false. However, if the list has 10, 20, 20, and 30, it will return true. If the list is empty or has only one number, the function should return true as well.
- `bool append(const List l);`  
This function will append the contents of the List l to the current list. If the combined list would contain too many elements, do not change the state of the list; just return false. Otherwise, copy the elements from l into the current list. Do not change l (you can't if it's marked const). HINT: You can use the existing append member function that takes an int *inside* this function.
- `bool remove(int value);`  
This removes the value given in an argument in the list. For example, if a list contains 10, 30, and 20, `remove(30)` will remove the value 30 from the list and return true. For the list 10, 20, and 20, `remove(40)` will not remove anything from the list and return false. If a list has duplication like 10, 30, 20, and 30, `remove(30)` should remove all 30s in the list and return true. So, the result list will have 10 and 20. You can use the existing *erase* member function *inside* this function.

#### Final Notes:

- Read the sample run carefully to understand the requirements of the problem correctly.
- This program should be uploaded to iLearn as hw03-2.cpp