**Program 4 – A Course Schedule Tracker**

The goals of program #4 are to become familiar with pointers and dynamically allocated arrays in the context of classes.

You have decided to make a simple “course schedule” program, that keeps track of the classes you are taking this term and the assignments to be done. You plan to keep at least the following information (you are welcome to tailor this to fit your needs):

Assignment class:

private attributes

* + - Assignment name must be a pointer to a dynamically   
      allocated array
    - Assignment due date a string

public methods

* + - Constructor that appropriately initializes the private  
      attributes. It will dynamically allocate an array based on the length of the assignment name read from the file and passed to the constructor.
    - Copy constructor because there is a pointer attribute in the class
    - Set and get methods for the attributes of the class
    - Overloaded assignment operator
    - Destructor that deallocates the assignment name and its pointer

Course class:

private attributes

* + - * Course Name a string
      * Course Number a string
      * Vector of Assignment objects

public methods

* Constructor that appropriately initializes the private attributes except for the vector which does not have to be initialized
* Set and get methods for course name and number
* Display method that displays a nicely formatted table containing the course name, number and all the assignments (including name and due date)
* Add Assignment method that receives an assignment object and adds it to the vector

**Input File**

You will create your course file to be in the following format:

* + First line: course name and number separated by commas
  + Second line: number of assignments for that course
  + Third line: the assignment name and the due date separated by a comma.
  + The Third line will repeat for each assignment until all assignments are listed for that course.
  + Then the first and second line will repeat for the next course , followed by all assignments for that course, etc. for all courses.

**Program requirements**

1. You must have at least the two classes identified above. Make sure to put the class interface in a .h file and the implementation of the class member methods in a .cpp file
2. The main driver will contain a vector of pointers to course objects. For every course’s data, it will dynamically create a course object. It will then create each assignment object from the data read on the following lines and add it to the course. The pointer will then be added to the vector of pointers of course objects.

Hint: When reading an assignment line, if you read the assignment name as a string, you must convert it to a char pointer using c\_str before passing it to the Assignment constructor.

1. After the entire file is read and the vector of pointers to course objects populated, call the display method on each course object.

**Grading Rubric**

Assignment class definition as explained above including separate header and implementation files. 25 pts

Course class definition as explained above including separate header and implementation files. 25 pts

Program is able to read in files of various sizes (i.e. # of courses and assignments) 20 pts

Correct usage of dynamic allocation of assignment name and the vector of pointers to course objects 20 pts

The main functionality as defined course requirements above 10pts

**Submission Details**

You are to submit a compressed .zip file of your Program 4 with the name LastName-Program4.zip. Make sure that when un-compressing your file, you see all source files that belong to your program, including a minimum of Assignment.h, Assignment.cpp, Course.h, Course.cpp, Program4.cpp and your data file. Due . April ;11th, 2022 @ 11:59PM. You may submit up to 48 hours late for a 30 point penalty.