CIS 236 – Programming in C

Program:	**Week 11**
Points:	**20**
Chapters:	**8**

Description

Write a program that prompts the user for course information, and then prints the courses in a neatly formatted list.

Learning Objectives

In this assignment, you will:

- Use lists of strings
- Use functions with string parameters
- Use string manipulation functions
- Use character classification functions

Requirements

Your code **must** use these functions, using these names (in addition to main):

1. getPrefix
2. getNumber
3. getName
4. combineStrings
5. printList

Requirements for the getPrefix Function

Purpose: This function reads the course prefix typed by the user and validates it.

Parameter: 1 string to hold the course prefix

Algorithm: Prompt the user for a course prefix as a string.

Validate the prefix according to these rules:

1. The prefix must contain exactly three letters.
2. The letters can be any combination of letters from A to Z.
3. You do not have check for the case (upper or lower or any combination is valid)

If the course prefix is not valid, display an error message and continue to prompt the user until a valid course prefix has been entered.

Return value: None

Requirements for the getNumber Function

Purpose: This function reads the course number typed by the user and validates it.

Parameter: 1 **string** to hold the course number

Algorithm: Prompt the user for the course number **as a string**.

Validate the prefix according to these rules:

1. The number must contain exactly three characters.
2. The characters must be the values '0' through '9'.
3. The characters can be in any combination or order.

If the course number is not valid, display an error message and continue to prompt the user until a valid course number has been entered.

Return value: None

Requirements for the getName Function

Purpose: This function reads the course name typed by the user and validates it.

Parameter: 1 string to hold the course name

Algorithm: Prompt the user for name of the course as a string that can contain spaces.

Validate the name according to these rules:

1. The course name can contain any character or digit (including spaces).
2. The course name cannot be longer than 16 characters.

If the course name is not valid, display a message indicating the string will be truncated, and add the null terminator as the last character in the string.

Return value: None

Requirements for the combineStrings Function

Purpose: This function creates one string from the string parameters passed to it.

Parameters: 4 strings – a string to hold the result, the course prefix, the course number, and the course name

Algorithm: Combine the course prefix, the course number, and the course name into one string, in this order: prefix, space, number, space, name. (Do not use commas in the code – they are just for informational purposes here.)

Return value: None

Requirements for the printList Function

Purpose:	This function prints the strings in the list.
Parameter:	List of strings
Algorithm:	Print the course information saved in the list.
Return value:	None

Requirements for main:

1. Declare a list of strings that represent course information. The maximum number of strings is 5. Each string can be at most a length of 25. Initialize the list to the null terminator.
2. Using a counter-controlled loop:
    a. Call the getPrefix function with a string argument.
    b. Call the getNumber function with a string argument.
    c. Call the getName function with a string argument.
    d. Call the combineStrings function with four string arguments.
    e. The result of the combineStrings function must be saved at the current index in the course list.
3. After the loop, call the printList function with 1 list argument.

Sample Run (Shows 6 courses – yours only needs to save 5)

```
Course prefix: CIS
Course number: 201
Course name: Programming

Course prefix: 201
Course prefix must be 3 characters
Course prefix: ENG
Course number: 202
Course name: English

Course prefix: MATH
Course prefix must be 3 characters
Course prefix: MAT
Course number: 203
Course name: Calculus

Course prefix: PHY
Course number: PHY
Course number must be 3 digits
Course number: 204
Course name: Physics
```

```
Course prefix: BIO
Course number: 25555
Course number must be 3 digits
Course number: 205
Course name: Biology

Course prefix: CIS
Course number: 206
Course name: Programming is Awesome
Course name exceeds 16 characters and will be truncated

Courses:
CIS 201 Programming
ENG 202 English
MAT 203 Calculus
PHY 204 Physics
BIO 205 Biology
CIS 206 Programming is A
```

Requirements for Full Credit on This Project

SUBMIT YOUR OWN WORK – Plagiarism is not tolerated in this course. Please review the section on the Academic Honor Code in the syllabus. I will not hesitate to drop you from this class if you submit a program that is plagiarized.

COMPLETE AND ACCURATE – Your program must compile, execute, and give accurate output.

FOLLOW ALL REQUIREMENTS ACCORDING TO THE INSTRUCTIONS – Follow the instructions as written for completing this project, even if you [think you] know a "better" way to do something.

COMMENTS – Include comments in your code. There must be a comment at the top of your program that includes your name, the program number, and a description of the program. There must be comments at each important step in your program that describes that step. Every variable must include a comment describing its purpose.

BEST PRACTICES – Follow best practices in C programming as discussed in class and in the textbook, including, but not limited to, appropriate use of white space, indenting, alignment, meaningful identifier names, etc. Points will be deducted for sloppy code that is hard to read, even if it works, so pay attention to these details.

SUBMIT ONLY .c SOURCE CODE – Pay attention to the file extension of the source code file you submit. I will deduct points for not following this requirement.

SUBMIT ALL FILES BEFORE THE DUE DATE – Submit your .c source code file to the dropbox for this assignment on Canvas before the due date. Do not submit executable files. Do not submit project files from an IDE. I will not accept links to online storage.