CSCE 315 Programming Studio

Individual Assignment

Fall 2017

Due Sunday, 9/17 11:59 p.m.
This assignment will count for 10% of your overall grade

You are to create a program that you use for keeping track of textbooks required and recommended for classes and the cost of those textbooks. All input will be taken from standard input. Each line of input will begin with a single or double character code that identifies the type of operation that line describes.

The possible line formats are:

Command	**B <ISBN> <Title>**
Explanation	Define a book. The ISBN is a 13-digit number. The Title is a string of arbitrary length (ending with a line break). (*Note that you favorite integer data type may not big enough for 13-digit numbers. In that case, it is OK to use a C++ std::string instead to store this number*)
Example	`B 1234567890123 Programming for Programmers`
Command	**D <ISBN> <A | E | D> <value>**
Explanation	Define a characteristic of a book (for the book with the given ISBN. If ‘A’ is used, then the Author is set, and the value is name of the author (string of arbitrary length, ending with end of line). If ‘E’ is used, the value is the number of the edition (a positive integer). If D is used, it is the date of publication, where the date is in MM/YYYY format (no day).
Example	`D 1234567890123 D 01/2017`
Command	**M <ISBN> <Cost> <N | U | R | E>**
Explanation	Define the cost for a book. ISBN is the ISBN number; Cost is an amount, expressed as a floating-point number with 2 spaces after the decimal point. The final letter indicates whether the price is for a New, Used, Rented, or Electronic version of the book. A book may have a different price for each format.
Example	`M 1234567890123 39.99 N`
Command	**C <Department Code> <Course Number> <Name>**
Explanation	Define a course. The Department Code is a 4-letter code for a department. The course number is a 3-digit number for the course. The name is a name for the course – a string of arbitrary length, ended by the end of the line.
Example	`C CSCE 315 Programming Studio`
Command	**A <ISBN> <Department Code> <Course Number> <Section Number> <R | O>**
Explanation	Assign a book to a class. The ISBN is the ISBN number for the book. The Department Code and Course number are as in the definition of a course. The section number is a 3 digit integer. The final digit is either R for required or O for optional
Example	`A 1234567890123 CSCE 315 501 R`

Command	**GC <Department Code> <Course Number>**
Explanation	Print the books required and optional for all sections of a given course
Command	**GS <Department Code> <Course Number> <Section Number>**
Explanation	Print the books required and optional for a given section of a course
Command	**GB <ISBN>**
Explanation	Print all information known about a particular book
Command	**PB**
Explanation	Print a list of all books that are defined
Command	**PC**
Explanation	Print a list of all courses that are defined
Command	**PY <MM/YYYY>**
Explanation	Print all books with known publication dates in the given month/year or later
Command	**PD <Department Code>**
Explanation	Print the list of all books used in a department, given by department code. Do not list by section
Command	**PM <Department Code>**
Explanation	Print the AVERAGE minimum and maximum costs of all books in a department. Minimum cost is the cost of the cheapest version of required books for a section. Maximum cost is the cost of the most expensive required and optional books in a course. If a book has no known costs, ignore it. The average is the average across all sections in a department for which there is some required or optional book

Notes:
- The program should be developed in C++
- Books will be defined before details or costs are added
- Books and courses will be defined before books are added to courses.
- Note that later lines can overwrite information from previous lines. For example, a price could be updated, an edition number could change, etc. The last of the input lines should be the one used.
- Printed lists should be reasonably informative, with all information provided possible (e.g. when printing a book, print the title, author (if known), costs (if known), etc.

Grading

As for grading, 35% of the grade will depend on how well the code works, and whether it provides the functionality described. The remaining 65% will be based on how well you have done with naming, style, and commenting in the code. Thus, pay particular attention as you write your code so that the names used throughout the program are good, the style is good, and the comments are good. Turn in will be to ecampus.

For grading, the naming/style/commenting will be graded using a peer review system. You will be required to perform peer review yourself, or your grade on your assignment will be a 0.

The grade on the correctness portion of the assignment will be determined by analyzing the output produced for sample sets of input data. Thus, you must not only process the data, but also produce reasonable outputs in response to commands that can demonstrate that you are indeed storing the data correctly. Code itself will not be analyzed for partial credit of correctness, only the output data.