

24-780 Engineering Computation

Problem Set 02

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas page. The file name of the ZIP file must be:

PS02-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS02-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Canvas page. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file **MUST** include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure your program can be compiled with no error in one of the compiler servers. Don't wait until the last minute. Compiler servers may get very busy minutes before the submission deadline!

The ZIP file needs to include:

- C++ source file of your program (PS2-1). The file name needs to be ps2-1.cpp
- C++ source file of your program (PS2-2). The file name needs to be ps2-2.cpp

Submission Due: Please see Canvas

START EARLY!

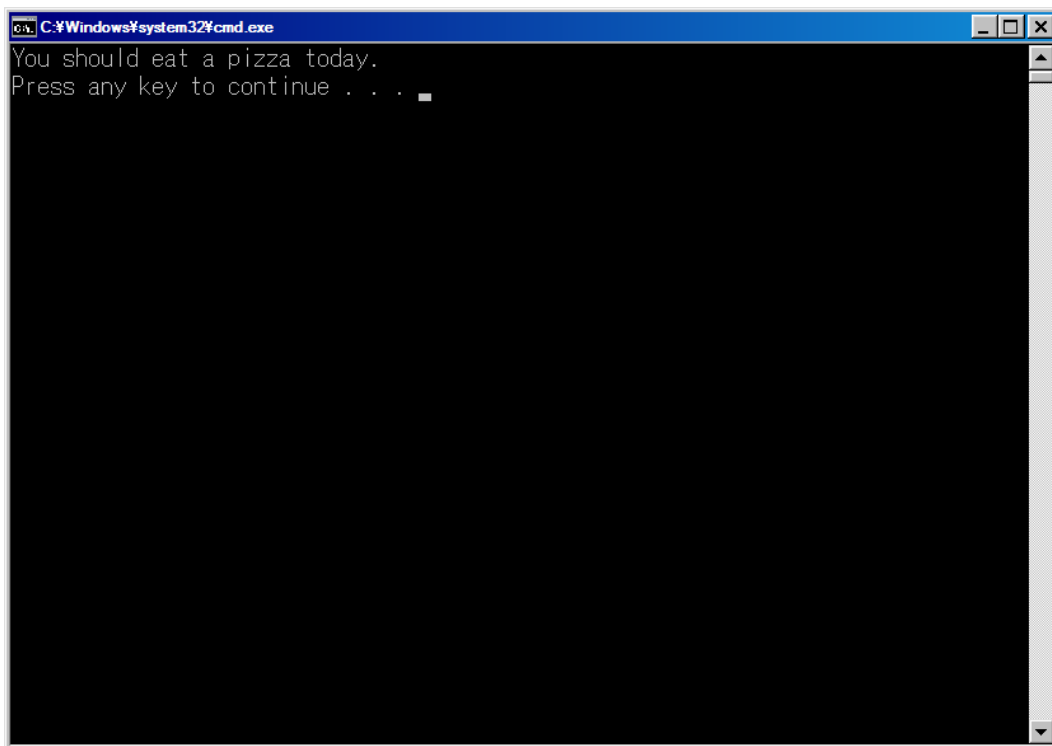
Unless you are already a good programmer, there is no way to finish the assignment overnight.

PS2-1 Deciding where to go for lunch [30 points]

Decisions, decisions! Every day, we need to make difficult decisions including what to eat for lunch. Famous physicist, Dr. Richard Feynman solved this problem by eating same thing for lunch every day. But, we need to eat balanced food to stay healthy. So, how about let the computer decide what to eat? Write a program that randomly tells you what to eat for lunch from: hamburger, pizza, sub, hot dog, doughnut, and pancake. Make sure:

- (1) The result is randomized based on the timer, and
- (2) Use switch and case statements.

When you compile your program and run, the program should print one line of message like:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\system32\cmd.exe". The window has a black background with white text. The text displayed is "You should eat a pizza today." followed by "Press any key to continue . . . " and a small cursor. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

The C++ source code (ps2-1.cpp) must be included in the Zip file that you submit to Canvas.

Make sure your program can be compiled with no error in one of the compiler servers. Don't wait until the last minute. Compiler servers may get very busy minutes before the submission deadline!

PS2-2 Digital Flashcards of Multiplication [70 points]

In this problem, you will write a C++ code that serves as a learning tool for elementary multiplication from 1×1 to 12×12 . Your C++ code should be a console application and roughly mimics the functionality of a conventional paper flashcard, shown in the figure below.



A flashcard is any of a set of cards bearing information, as words or numbers, on either or both sides, used in classroom drills or in private study. Flashcards can bear vocabulary, historical dates, formulas or any subject matter that can be learned via a question and answer format. Flashcards are widely used as a learning drill to aid memorization by way of spaced repetition.

Your C++ console flash card should:

- First ask the user how many flash cards to work on. The number needs to be between 1 and 144. If the user requests less than 1 or greater than 144 flash cards, show a message (eg. "The number of cards must be between 1 and 144.") and prompt the user to re-enter a different number.
- Show questions one by one on the console window and prompt the user to type the answer. If the user types a wrong answer, tell the user that the answer is wrong and show the correct answer then move on to the next card. If the answer is correct, move on to the next card.
- When all the flash cards are answered, display a feedback comment that includes:
 - The number of problems,
 - Time required to complete the problems in the number of seconds, and
 - The number of correct answers and percent that the user answered correctly.

For example,

You answered 25 problems in 80 seconds.

You answered 20 problems correctly (80%).

- Problems should be selected and presented randomly, but no same question should appear more than once.

Write the C++ program (named ps2-2.cpp). Make shuffling a separate function. The C++ source code must be included in the Zip file that you submit to Canvas.

Make sure your program can be compiled with no error (no red lines) in one of the compiler servers.
Don't wait until the last minute. Compiler servers may get very busy minutes before the submission deadline!