Name:

|  |  |  |
| --- | --- | --- |
| Problem | Points | Score |
| 1 | 50 |  |
| 2 | 50 |  |
| Total | 100 |  |

Notes:

1. For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.
2. The first step in this exam is to create a workspace in the following directory:

/data/courses/ece\_1111/current/exams/ex\_03/lastname\_firstname

1. Set the permissions using “chmod -R u+rwx,g-rwx,o-rwx <lastname\_firstname>” so only you have read and write permission to this directory. Create two subdirectories within this directory: p01 and p02. Put ALL your code in these directories. Do not touch your files after the exam is over.

As explained in class, for this exam, you need to follow our standard approach to writing C++ code. Your solution should include a make file, a header file containing your class definition, a driver program (p01.cc for problem 1) and an implementation file. Data in your classes should be “protected”. Member functions should be public. You do not need any private functions for this exam.

You must complete problem no. 1 before you can proceed to problem no. 2. Problem no. 1 should be put into a directory /p01. When you proceed to problem no. 2, create a directory /p01\_v02, copy your code from /p01 and extend it as requested. To save you time, you do not need to rename p01 for problem no. 2.

**Problem No. 1 (50 pts)**: In this problem, we are going to create a class that converts the days of the week to a number. Create a driver program that takes the names of a day of the week as input:

ece-000\_[1]: p01.exe Tuesday Wednesday

day1: tuesday is day 2 of the week

day2: wednesday is day 3 of the week

Your class should be called Convert and needs to include a table that maps the day of the week to a number (“Sunday” = 0, “Monday” = 1, …, “Saturday” = 6). You have some freedom in how you implement this, but the table needs to be declared as static data so there is only one copy of it for all instances of the Convert class.

The Convert class should have a method named “convert” that accepts a character string as an argument and returns an integer containing the number. This method must use a table lookup – it cannot use if/else if logic. Your driver program should print out the numeric value for argv[1] and argv[2] as shown above using the objects day1 and day2.

Your program must be case-insensitive – argv[1] and argv[2] can be specified using mixed case.

**Problem No. 2 (50 pts)**: Create a second class called Calendar that includes a Convert object, named convert\_d, as protected data. This class should have a method named “sum” that returns the sum of the numeric values for days of the week. This method takes an array of character strings as input, and returns an integer containing the sum of the numeric values corresponding to the days specified. For example:

ece-000\_[1]: p02.exe Tuesday Wednesday

sum = 5

The driver program passes an array of strings to the Calendar object, calls its sum method, and prints the sum of the corresponding days.

**Additional Notes:** You will be graded on how clean and well-designed your solution is. Try to keep your code as simple and short as possible. Delete unnecessary code.