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_02/lastname_firstname

(3) 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. You will use these for problems 1 and 2 respectively. Put ALL your code in these directories. Do not touch your files after the exam is over.

(4) You must use a make file, a header file and a main program file named p01.cc (or p02.cc). All other code needs to go into an implementation file called p01_00.cc (or p02_00.cc).

Problem No. 1: This binary file:

/data/courses/ece_1111/current/exams/ex_02/picone_joseph/p01.dat

contains three numbers, each of which has a value of 27. Unfortunately, you don't know what data types were used to write the data. Figure out what the data types are and write a program, p01.cc, to read the file and print its values. Your printout should look like this:

```
1: <type> value
2: <type> value
3: <type> value
```

For example, if you determine that the first number is a double, the second number is a float, and third number is a char, your output will look like this:

```
ece-000_[1]: p01.exe /data/courses/ece_1111/current/exams/ex_02/picone_joseph/p01.dat
1: double 27.0
2: float 27.0
3: char 27
```

Big Hint: Use the od command to figure out what is in the file.

Problem No. 2: In HW #7, we learned how to read a file using frames and windows. Modify your program so that it reads a binary file of short integers using a frame size of M samples and a window size of W samples. Your interface must be the following:

```
p02.exe <filename> M W
```

Use this file for testing:

/data/courses/ece_1111/current/exams/exam_02/picone/p02.dat

This file contains short integers ranging in value from 1 to 20:

```
nedc027_[1]: od -s /data/courses/ece_1111/current/exams/exam_02/picone/p02.dat
0000000      1     2     3     4     5     6     7     8
0000020      9    10    11    12    13    14    15    16
0000040     17    18    19    20
0000050
```

There are two things your program must do differently:

(1) The window, instead of being center-aligned, must be left-aligned. This means the first window should be the first W samples in the file. The second window must be this window shifted by M samples. The third window must be the second window shifted by M samples. For example, for p1.dat for M =2 and W = 4:

```
frame 1: [1 2 3 4]
frame 2: [3 4 5 6]
frame 3: [5 6 7 8]
```

Your code must work for all combinations of M and W and different file lengths, and must handle the end of file condition properly (values beyond the end of file are assumed to be zero). You must handle the case where W < M.

You must only read M samples with each iteration after initialization. You cannot simply position the file pointer and read W samples each time. The total number of samples you read must be equal to M so that you are not doing unnecessary I/O.

(2) You must multiply each window of data, s_w, and a weighting function, h, where h is a vector of length W whose values are all 2. Then you must sum the squares of the result to compute “energy”. Print the following information to stdout: input filename, M, W, and for each frame, the frame index and the energy value.

In the previous example, the output will be (after the weighing function is applied):

M =2 and W = 4:
frame 1: 120 $[(1*2)^2+(2*2)^2+(3*2)^2+(4*2)^2\ ]$
frame 2: 344 $[(3*2)^2+(4*2)^2+(5*2)^2+(6*2)^2\ ]$…