Project 7 (30 points): If statements
➔ 10% bonus if your program output matches the sample solution exactly ←
Submit Your Solution Using ANGEL by Noon on Friday, 2/22/13
(A late submission drop box will be available on 02/22/13 from Noon to 2pm)

Obtaining Project 7 Input Order and Output Requirements:

(Sample Solution and Comparison Script)

To view the order and style of the input and output information for the project, run the provided solution by **typing the following at a terminal window prompt.** (**Note**: Input files must be present in the directory that the terminal window is currently in.)

Sample Solution – for determining vertical spacing differences

/home/work/cpe112/Executables/Project_07/Project_07_solution
Provided sample input files are zipped in the file P7_in.zip
Be sure to read the README.txt file included with these zipped input files

Comparison Script – for determining line differences

/home/work/cpe112data/Project_07/CompareSolution.bash Project_07.cpp

Project 7 Restrictions

Only material from Chapters 1 through 5 and any extra code in this handout is allowed.
You cannot use any C++ techniques that are covered in Chapters 6 and higher.

You are not allowed to use any global variables. Global variables are declared above the **int main()** line in a program. Global constants can be used

Starting Project 7:

- Open a terminal window and move (cd) into the Project_07 directory created in the CPE112_SPR13 directory (This is the directory structure created in project 1.) The command for this is: **cd CPE112_SPR13/Project_07 (typing cd ~/CPE112_SPR13/Project_07 works as well)**. You will need to modify the names as necessary to match your capitalization style for the two directories.

- Download all needed files from ANGEL into this directory. Using your favorite text editor (i.e. nedit or gedit), open the existing **Project_07.cpp** file (this is the header file created in Project 2) and add code to it to complete this project. Once you have finished with the program and it compiles without syntax errors, run the executable and verify that the output for your program matches the output from the provided solution executable.

- Remember to use the firefox browser when viewing ANGEL. Also, to view a pdf file saved in a directory, type the command **acroread filename.pdf** at the prompt in a terminal window that has the same working directory as the directory in which the file is present.

- **Once you are satisfied with your solution, submit Program_07.cpp via Angel**.

NOTE: make sure that you do not change the order in which the information is entered. An automatic script is used to process all lab submissions, and if the order of the input information is modified, the script will not work properly with your program.

Project 7 Description

This program requires the utilization of concepts in Chapter 4 and Chapter 5. This program is to calculate the cost of mailing a package. The cost of the mailing is based upon three values: the base rate for mailing the package, the length adjustment factor and the weight adjustment factor. The sum of the dimensions (height, length and width) of the package is used to determine the length adjustment factor. The weight of the package is used to determine the weight adjustment factor.

There are three types of packages that can be mailed. These packages are shown below along with their base rate for shipping:

Package type	Base Rate
parcel	**$2.25**
media	**$3.50**
bulk	**$4.75**

Note: for this program the package types are written with all lowercase letters.

The value used for determining the length adjustment factor is the sum of all three dimensions of the package (total length = width+height+length). The three possible adjustment factors for length are:

Total Length in inches	Length Adjustment Factor
Less than 40 inches	**1.00**
Greater than or equal to 40 inches but less than 60 inches	**2.50**
Greater than or equal to 60 inches	**3.75**

The weight adjustment factor is based on the weight of the package only. The three possible weight adjustment factors are:

Weight in pounds	Weight Adjustment Factor
Less than 10 Pounds	**1.00**
Greater than or equal to 10 pounds but less than 25 pounds	**3.00**
Greater than or equal to 25 pounds	**4.50**

Once the base rate, length adjustment factor and weight adjustment factor have been determined, the cost (in dollars) of mailing the package is determined by multiplying all three values together.

Cost = (Base Rate) * (Length Adjustment Factor) * (Weight Adjustment Factor)

All package information is read from a user specified input file and all output information is written to the terminal. Run the provided sample solution to see what information needs to be written to the terminal.

Project 7 Sample Input File

The information in the input file is as shown below.
All package dimensions and weight are to be read as integer values.
Comments shown are for informational purposes and are not present in the input file.

```
Ron Bowman Project 07 input file    // line 1 is the header info line
parcel                              // line 2 is the package type – parcel, media or bulk (all lowercase)
10                                  // line 3 is the package length in inches
20                                  // line 4 is the package width in inches
30                                  // line 5 is the package height in inches
25                                  // line 6 is the package weight in pounds
```

The first action taken by the program is to prompt the user for the name of the input file. Once that name has been read into a string variable, echo print (use a cout statement to output the name fo the file just entered by the user) the name to the screen. Then use the open function to associate the name of the file entered with an input file stream variable. **After the input file has been opened, verify that the input file stream is valid.** If the input file stream is not valid, then the input file does not exist, and the program terminates after displaying an appropriate message.

Reading Information from the Input File

- the header line is read and then ignored – it is not needed
- Then the package type is read.
- Next, length, width, height and weight – **all integer values** - are read (in that order) from the file.

Input is to follow the same order as that illustrated by the provided solution executable. Output is to match that shown by the provided solution.

Several problems may exist within the data in the input files. The table below states the possible problem and what the program response should be. It also indicates the order in which the testing is to be performed.

Problem	Action #1	Action #2	Input file
Input file does not open	Print out error message	Terminate the program	Not/a/file.txt (or P7_in10.txt)
Invalid Package type	Print out error message	Terminate the program	P7_in4.txt
Invalid Character read when reading the package parameters	Print out error message	Terminate the program	P7_in5.txt
One or more package parameters is less than 0	Print out error message.	Terminate the program	P7_in6.txt - P7_in9.txt

<Project 7 Hints>

- Several header files are required: **string, iomanip, and fstream**.
- **The base rate, total cost and all adjustment factors are float values.**
- **The package characteristics - length width height and weight are all integer values**

- Remember all dimensional values are integer values and the weight is an integer as well. The name of the type of package is written with all lower case letters.
- Make sure that the break points for the adjustment factors are tested properly. For example using a value of 10 pounds for the weight will verify that the program is using the correct weight adjustment.
- Use three different if-then-else-if statements to determine the total cost.
    - One set of statements determines the base rate of the package,
    - another set of statements determines the length factor and
    - the last set of statements determines the weight factor.

- Use the setfill operator and setw to place the periods between identifying phrases and the value associated with that phrase.
- The dashed lines in the output are 50 dashes long
- Row of asterisks at bottom of all error messages is 41 *'s
- The identifying phrases in the output are in a field width of 30.

- Read the README.TXT file included with the input files. Look at the input files. Run your program with all provided input files