Project 1, Program Design

A furniture renting store rents several types of furniture to customers. It charges a minimum fee for the first month. The store charges an additional fee every month in excess of the first month. There is a maximum charge for any given year. Write a program that calculates and prints the charge for a furniture rental.

Furniture Piece	First Month (minimum)	Monthly Cost after the First Month	Maximum per Year
Sofa	$20	$12	$100
Loveseat	$15	$10	$80
4 Post Bed	$25	$8	$105
Dresser	$10	$5	$50
Kitchen Table	$25	$10	$120

An example input/output:

```
Please select from the following menu: 1. Sofa, 2.
Loveseat, 3. 4 Post Bed, 4. Dresser 5. Kitchen Table

Enter furniture selection: 1
Enter months rented: 2
Amount due ($): 32
```

Requirements:

1. Name your program *furniture.c*
2. The user enters the furniture selection, enter the number of months rented for a customer. The program calculates and prints the charge.
3. Your program should validate the furniture selection. If the selection is not in the range of 1 to 5, print a message and exit the program.
4. Your program should validate the months rented. If the number entered is less than 1, print a message and exit the program.

Before you submit:

1. Compile with –Wall. –Wall shows the warnings by the compiler. Be sure it compiles on ***student cluster (sc.rc.usf.edu)*** with no errors and no warnings. All projects are graded on the student cluster.

   gcc –Wall furniture.c

2. Be sure your Unix source file is read & write protected. Change Unix file permission on Unix:

chmod 600 furniture.c

3. Test your program with the shell script on Unix:

   chmod +x try_furniture
   ./try_furniture

4. Download the program *furniture.c* from student cluster and submit it on Canvas>Assignments.

Grading

Total points: 100

1. A program that does not compile will result in a zero.
2. Runtime error and compilation warning 5%
3. Commenting and style 15%
4. Functionality 80%

Programming Style Guidelines

The major purpose of programming style guidelines is to make programs easy to read and understand. Good programming style helps make it possible for a person knowledgeable in the application area to quickly read a program and understand how it works.

1. Your program should begin with a comment that briefly summarizes what it does. This comment should also include your **name**.
2. In most cases, a function should have a brief comment above its definition describing what it does. Other than that, comments should be written only *needed* in order for a reader to understand what is happening.
3. Variable names and function names should be sufficiently descriptive that a knowledgeable reader can easily understand what the variable means and what the function does. If this is not possible, comments should be added to make the meaning clear.
4. Use consistent indentation to emphasize block structure.
5. Full line comments inside function bodies should conform to the indentation of the code where they appear.
6. Macro definitions (#define) should be used for defining symbolic names for numeric constants. For example: `#define PI 3.141592`
7. Use names of moderate length for variables. Most names should be between 2 and 12 letters long.
8. Use underscores to make compound names easier to read: `tot_vol` or `total_volumn` is clearer than totalvolumn.