CS162 Project 1 Budget Control in Shopping Cart

Do-Not List for All Projects in CS162:

- No Global Variables (you can have global constants)
- No use of the stdio library (use iostream and fstream)
- Instead of the string class, you will be using arrays of characters and the cstring library

Things You Should Do:

- Follow the style guide for this class
- Your programs should always guard against bad data being entered by mistake
- You must follow all specifications for this project and not deviate from it.

Goals for This Project:

- Simple data modeling
- Data input/output and data validation
- Flow control: if/else and loops

Problem You Need to Solve for This Project:

Have you ever been low on cash and couldn't go beyond a certain dollar limit when shopping? You sort of need a calculator in your head. It would be cool if a device was actually part of the cart and as you add an item into the cart it would increment your total counter. To solve this, we are going to write a program that keeps a tally of the amount that we have spent so far as we visit a store.

What Your Program Should Do:

1. Allow the shopper (user) to enter in the product name and the cost. This should be echoed and confirmed. Make sure to check for bad data.
2. Output the total thus far.
3. The user should be allowed to continue this until they want to checkout.
4. Your program needs to keep a running total.
5. Upon checkout, the grand total should be displayed.
6. Display money using proper formatting. You will need to have the following before displaying dollars and cents:

```
cout.setf(ios::fixed,
ios::floatfield);cout.setf(ios::showpoint);
cout.precision(2);
```

Grading Requirements:

The goals for this project say “Simple Data Modeling”. There is no need to use any structs, or classes, or pointers for this assignment. It should be very simple, should use char arrays, loops, and keep a running total of the amount. The menu should be displayed in a loop until the user enters ‘q’ to quit. Failure to follow any of these specifications will result in deduction of points.

How To Submit Your Work:

Organize your directories:

mkdir cs162 //create directory for cs162

cd cs162   //go inside cs162

mkdir project1 project2 project3 project4 project5 //create directories for each project

cd project1   //go inside project1 directory

```
vi project1.cpp //create source file for the project

g++ -o proj1 project1.cpp //compile the project

./proj1                    //run the project
```

Email the .tar file containing your project directory to your instructor on the linux server. To create a tar file of the project directory:

- remove the executable files in the project directory: rm proj1
- go to the parent directory that contains the project directory: cd ..
- tar -cvf project1.tar project1

To email the tar file to the instructor and yourself:

```
mailx -s "cs162 project1 submission" -a project1.tar gd.iyer your-login-name
```

entering message here (an example: known bugs: ...)

```
Known bugs: ...

.
```

a single period at the beginning of new line will send the email

To check if you have sent the file correctly:

```
mailx
```

assuming the message index is 3, type

```
w 3
```

to save the message. It will ask you for attached file name if the message is not empty. If it's an empty message, 3 is the tar file name. The file will be saved in the current working directory.

The following command will extract from the tar file.

```
tar xvf tarFileName
```