Spring 2017 CMSC140 Programming Project 4: Expenses

Concepts tested in this project

- To work with Variables and Literals
- To learn and use different Data Type
- To learn and use Programming Style
- To work with cin command to get input
- To use conditional statements
- To learn and use loops
- To learn and use writing to files

Project Description

Write a program that lets the user enter his/her expenses for a specific category(for example Grocery, Clothing, Utilities, …) for 4 months, finds the total of expenses in that category as well as the total expenses in all the categories.

The negative amount of the expense is not allowed and the user will be asked to enter a positive amount for the expense as long as the input is not valid.

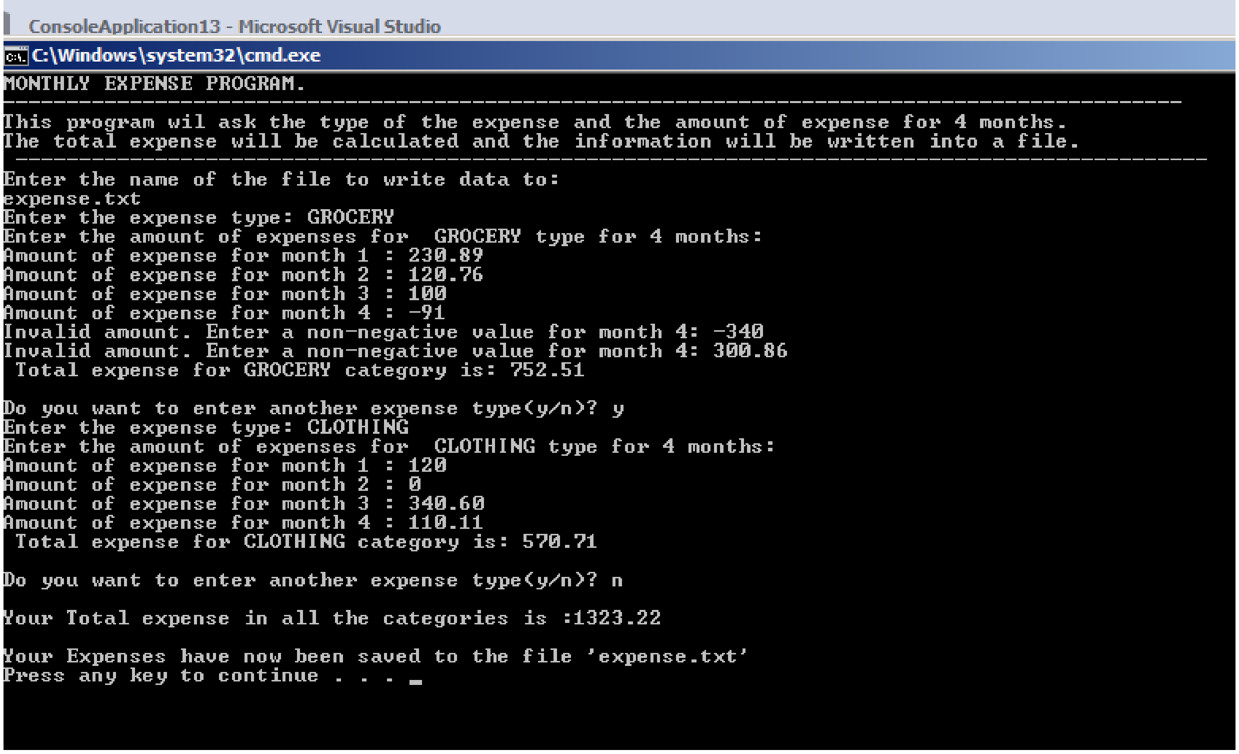
The information will also be written into a file, where the user specifies the name of the file**.**

Write, compile and run a C++ program that performs the specified tasks explained above depending on user's choice. **Refer to the screen shot of the sample output for more details**.

Create a pseudo code of the program before you start coding to observe the flow of the program.

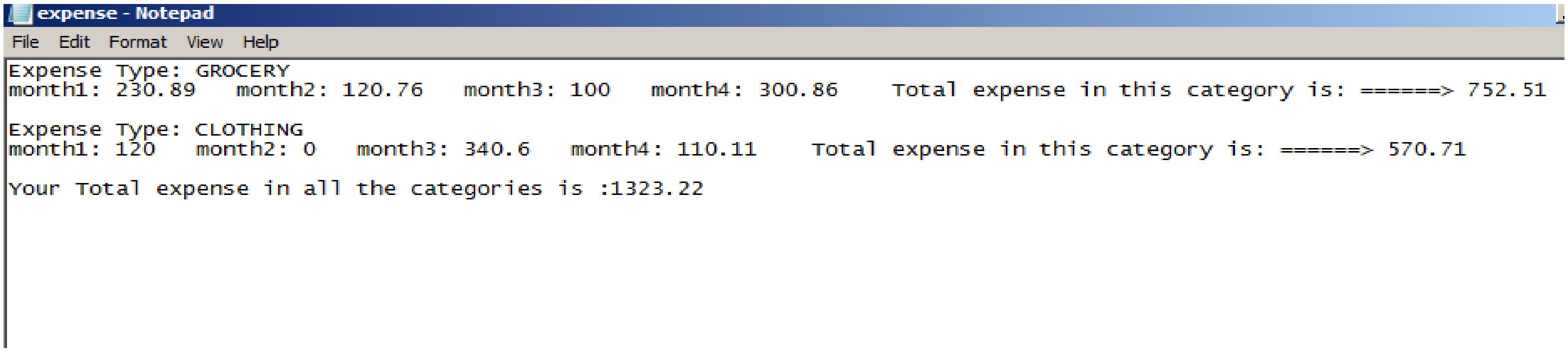
Spring 2017 CMSC140 Programming Project 4: Expenses

Following is a sample run of the program:


ConsoleApplication13 - Microsoft Visual Studio
C:\Windows\system32\cmd.exe

```
MONTHLY EXPENSE PROGRAM.
---------------------------------------------------------------------------------------------
This program wil ask the type of the expense and the amount of expense for 4 months.
The total expense will be calculated and the information will be written into a file.
-----------------------------------------------------------------------------------------------
Enter the name of the file to write data to:
expense.txt
Enter the expense type: GROCERY
Enter the amount of expenses for  GROCERY type for 4 months:
Amount of expense for month 1 : 230.89
Amount of expense for month 2 : 120.76
Amount of expense for month 3 : 100
Amount of expense for month 4 : -91
Invalid amount. Enter a non-negative value for month 4: -340
Invalid amount. Enter a non-negative value for month 4: 300.86
 Total expense for GROCERY category is: 752.51

Do you want to enter another expense type(y/n)? y
Enter the expense type: CLOTHING
Enter the amount of expenses for  CLOTHING type for 4 months:
Amount of expense for month 1 : 120
Amount of expense for month 2 : 0
Amount of expense for month 3 : 340.60
Amount of expense for month 4 : 110.11
 Total expense for CLOTHING category is: 570.71

Do you want to enter another expense type(y/n)? n

Your Total expense in all the categories is :1323.22

Your Expenses have now been saved to the file 'expense.txt'
Press any key to continue . . .
```

Following is the file created by the program for the above information:



```
Expense Type: GROCERY
month1: 230.89    month2: 120.76    month3: 100    month4: 300.86     Total expense in this category is: ======> 752.51

Expense Type: CLOTHING
month1: 120    month2: 0    month3: 340.6    month4: 110.11     Total expense in this category is: ======> 570.71

Your Total expense in all the categories is :1323.22
```

Project 4 Submission requirements:

Notes:

- Proper naming conventions: All constants, except 0 and 1, should be named. Constant names should be all upper-case, variable names should use "camel case" (i.e. start with lower case, with subsequent words starting with upper case: *hours**W**orked* for example) or underscores to separate words (i.e. items_ordered) (textbook, page 42)
- Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided.
- Documentation: The documentation requirement for all programming projects is one block comment at the top of the program containing the course name and CRN, the project number, your name, project description, the due date and platform/compiler that you used to develop the project. If you use any code or specific algorithms that you did not create, a reference to its source should be made in the appropriate comment block. Additional comments should be provided as necessary to clarify the program.
- Indentation: It must be consistent throughout the program and must reflect the control structure.
- **Program Header**: You should include one block comment (header) at the top of each program containing the course name and CRN, Instructor's name, the project number, your name, the date and a short description of the project as follows:

```
/*
 * Class: CMSC140 CRN
 * Instructor:
*  Project [number]
 * Description: (Give a brief description for Project1)
 * Due Date:
  * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.
   Print your Name here: __________
*/
```

Deliverables:

1. A Word document that includes:
   - Title Page with the following information
     - Project <#>, Due date (including year) , Your name, class, and section
   - Screenshots of the program
   - Source code of program
   - Pseudocode or Flowchart for the program
   - Highlights of your learning experience

2. Your source code (.cpp file). Your source code file should include a block comment (header) listed below.
3. The C++ files zipped and saved as LastNameFirstName_Project3_Moss.zip

   This .zip will not have any folders in it – only .cpp files.

   Note: This format is required to check for duplicate submissions using "MOSS" Plagiarism Detection Software.

Submit your completed assignment to **Blackboard** no later than the due date.

Grading Criteria for Project 4

This project will be graded using the following are components. **If program does not compile, project will get grade “0”**. Contact your instructor prior to the project submission due date, if you have compilation issues.

Attributes	**Value** (points)
Functionality (If project does not compile, project will get grade “0”)	Total 100
Displays the user choices appropriately	10
The required information described in project description, are asked from the user	15
Input validation check for the amount of expense	5
Program continuously asks for different categories as long user chooses continue.	10
Total expense calculated on each category	5
Total expense is calculated on all the categories	5
Information are written into file	10
Program executes correctly (produce expected output)	15
Meets all requirements	15
Overall Look-and-Feel	10
Total	Total 100 points

Project General Requirements (points will be deducted)

Attributes	**Value(points)**
Programming Style and proper naming convention: (see coding standards)	(-20 pts maximum)
Constants not all caps	-5
Curt or unclear variable names	-5
Long variable names should use camel case or underscores to separate	-5

words	
Comments and internal notes	
Sparse and inadequate comments.	-5
File header is not included	-5
Essentially no comments	-10
Indentation and white spaces should be a visual aid to understanding code structure	
Indenting is mostly okay, but sometimes inconsistent.	-5
No indenting, or very inconsistent indenting that is a barrier to understanding the code	-10
Lack of white space separating variables and operators. Lack of white spaces separating functions and major code blocks (later projects only)	-5
Test Plan	(-20 pts maximum)
Missing Entirely	-20
Cursory or inadequate testing	-10
Adequate overall, but missing a few crucial tests	-5
Missing Required Items (only if required for the project)	(-20 pts maximum)
Pseudocode, Flowcharts, or Hierarchy chart missing	-20
Screen shots cursory or incomplete	-5
Screen shots completely missing	-10
List of assumptions made (not applicable for Project 4)	-5
Highlights of your learning experience	-5
Awkward Code Internal Structure	(-10 pts maximum)
Hard-coding input values	-10
Poorly structured programming: inappropriate loop choices, Incorrect use of break statements to exit loops, and so on.	-5 each
Excessive reliance on global variables (e.g., using them to avoid pass by reference) (not applicable for Project 4)	-5
Processing array contents piecemeal rather than using loops (not applicable for Project 4)	-5
Other poor coding practices not mentioned	-5