CS 1 Introduction to Programming

SRC Assignment: Student Budgets

What is SRC?

The “Socially Responsible Computing” assignments are designed to introduce ethics and social impact topics broadly to students so that students are familiar with these concepts when you are eventually faced with ethical design decisions further down your CS journey.

Learning Objectives:

TBD

Goals:

In this assignment, you will learn to format output so that data is more easily understood by users of your program. For example, money should be displayed with a currency symbol, such as a dollar sign, “$”, a decimal, and appropriate number of digits to show cents.

You will also learn why it is important to be aware of expenses, especially as a new college student.

Programming Language and environment: Java, text-based input and output on the console.

Pre-Reading:

Being responsible with your money can be a big challenge when you start school. There can be many new expenses, related to rent, commuting to school, textbooks, tuition, and groceries. There can also be new sources of income, such as on-campus or off-campus jobs, scholarships, and internships. With school work, jobs, and other responsibilities, it can be overwhelming to keep track of everything.

If students lose track of income and spending, they can run out of money before more money comes in. This can lead to an inability to pay for living expenses like utility bills, rent, and food. It might also lead to missed payments for credit cards, tuition, or vehicles. Besides going hungry, other consequences can be a reduction

of your credit score due to missed payments. A low credit score can impact your ability to rent a place to live. A low credit score can make borrowing money for cars, homes, or high value items either more expensive or impossible.

https://www.wellsfargo.com/goals-going-to-college/student-budget/

To avoid these negative consequences, Wells Fargo, and other financial organizations have recommendations for college students. Some of these recommendations include:

1. Track your spending: keep a log of your expenses for a month or two to learn about what you spend your money on.
2. Make list of your income and expenses: Make a list of where you are spending money and where you have money coming in.
3. Do the math: After you total your income and expenses, determine how much you need for each semester and how much you have left over. This is a budget.
4. Revisit and adjust: As your income and expenses change, you should update your income, expenses, and your budget.

In this assignment, you will design and build a program that can collect budget items and format and display the budget data in a way that is easy for users to understand. This type of tool can help with recommendations 1 and 2.

Questions and Discussions Led by the Instructor:

1. What kind of expenses have you encountered as a college student?
2. What kind of income sources are available to you as a college student?
3. What are ways that you can keep track of income and expenses?
    a. Paper logbooks? Spreadsheets or other software or apps?
4. How can we differentiate income and expenses? Which should use a negative number?
5. When you write down your income and expenses, can it help to format the information neatly? What might happen if the information is laid out in a way that is confusing or difficult to read?
    a. The instructor can provide examples of confusing output, with issues in formatting or layout.

b. Periods are used in the United States to denote decimal monetary values, while commas may be used in other countries. Could this be confusing?

Test Cases:

It is helpful to have some examples of test cases for students to try out. Students, when they are early in their learning careers, often have difficulty coming up with good test cases for their programs.

Well-Behaved Data Examples: 5.22, -10.01, 0.05, -1.01, 7.42, -10.50, 5000.10, -2010.20

Slightly Troublesome Data Examples: 10.00, 10000, -2.000, -50, 0.1

Even More Troublesome Data Examples: 5.001, -21.338, -21.333.

Programming Assignments:

Programming Task 1: Formatting Data

Write a program that accomplishes the following:

- Declares an array of floats.
- Initializes the floats to the Well-Behaved Data Examples given above.
- Prints out a list in which the floats are printed in a well formatted fashion. See the example below. (Negative monetary values are usually shown with parentheses in spreadsheets and other financial software.) Format the data so that values with four digits for dollars and two digits for cents can be lined up.

```
$     5.22
$ (  10.01)
$     0.05
$ (   1.01)
$     7.42
$ (  10.50)
$  5000.10
$ (2010.20)
```

Programming Task 2: Inputting Data

Write a program that accomplishes the following:

- Has an input loop that repeated asks the user to enter the name of an income item or expense item.
- After taking the name, the program will then ask the user to enter the value of the item. Provide the user with instructions (i.e. no dollar sign, a decimal if needed, cents if needed). An example of how the data should be entered should also be provided (eg. Examples of valid input: -40.40 or 0.30).
- If the name of the item is blank, then the loop will terminate.
- When the loop terminates, the program will print the complete list of income and expenses. Each item's name will be listed with the corresponding monetary value on the same line.
- The monetary values will be displayed in a well formatted way. See the example below. Format the data so that values with four digits for dollars and two digits for cents can be lined up. (Negative monetary values are usually shown with parentheses in spreadsheets and other financial software.)

```
Return       $     5.22
Lunch        $ (  10.01)
Gumball      $     0.05
Drink        $ (   1.01)
Refund       $     7.42
Dinner       $ (  10.50)
Paycheck     $  5000.10
Rent         $ (2010.20)
```

Programming Task 3: Filtering Data

Write a program that builds upon Programming Task 2, but adds the following:

- Prints all the income items (positive values) first and provides an income subtotal.
- Prints all the expense items (negative values) second and provides an expense subtotal.
- Prints the sum of the income and expense subtotals. This may be negative or positive. If it is negative, you will need to display the negative sign and the paratheses.

- The monetary values, subtotals, and totals will be displayed in a well formatted way. See the example below. Format the data so that values with four digits for dollars and two digits for cents can be lined up.
- Test with the troublesome data examples. You may need to reject data that does not make sense as dollars and cents. You should let users know what the problem is and give them another chance to enter the data.

```
Income:
Return        $     5.22
Gumball       $     0.05
Refund        $     7.42
Paycheck      $  5000.10
Subtotal:     $  5012.79

Expenses:
Lunch         $ (  10.01)
Drink         $ (   1.01)
Dinner        $ (  10.50)
Rent          $ (2010.20)
Subtotal:     $ (2031.72)

Total:        $  2981.07
```

Reflection Questions

1. Do think it is helpful or not helpful for users to see budget data that is formatted and aligned?
2. Is it important for users to be able to easily read and understand their finances? Why?
3. What might happen if the financial data are not formatted and aligned?

Extra Credit

- The data was formatted such that monetary values are expected to be between -$9999.99 and $9999.99. Can you write code that would examine

all the values and adjust the formatting to be just wide enough so all the data lines up with the most extreme value in the data?