Nicholas Iudiciani & Altamas Kadawala

Business Applications Development

Prof. Hongwei Zhu

Assignment 1 – Annual Expense Estimator:

A Java program that records items and expenses for a year

**Table of Contents**

Overview……………………………………………………………

Design and Implementation……………………………….

Testing……………………………………………………………….

Discussion………………………………………………………….

Source Code………………………………………………………

**Overview**

For this first assignment, we were instructed to develop a Java program in Eclipse that would serve as a record for our annual expenses. We would record the items themselves, how often we would purchase these items on a daily basis for the whole year, their individual costs per item, and their annual total expense. The overall goal here is to show how much we spent that year to offer possible insights into our spending habits as college students.

For the input, my partner and I chose 8 different items that we felt are pivotal to our daily and educational lives. We calculated their respective amounts and implemented the proper code to lay out exactly what we wanted to show in the way we wanted. We chose to use a printf() command for each element of the list – the header, the actual data, and the final grand total amount – to show a clean out and block of code that seemed to flow nicely when we could just copy and paste the previous lines to input the code quicker. There are block comments and line comments within the code itself to explain why we chose the parameters for the output.

As for the output itself, we ran the program and were given an output record like the one illustrated in the assignment explanation, so we felt we had done it right. We looked back on the Quiz 2 Colleges code to see a similar output done with different input. The code that we presented was done in about 45 minutes, and we tried to model it after the code from Quiz 2 but had trouble deciphering it, which led us to produce this code instead.

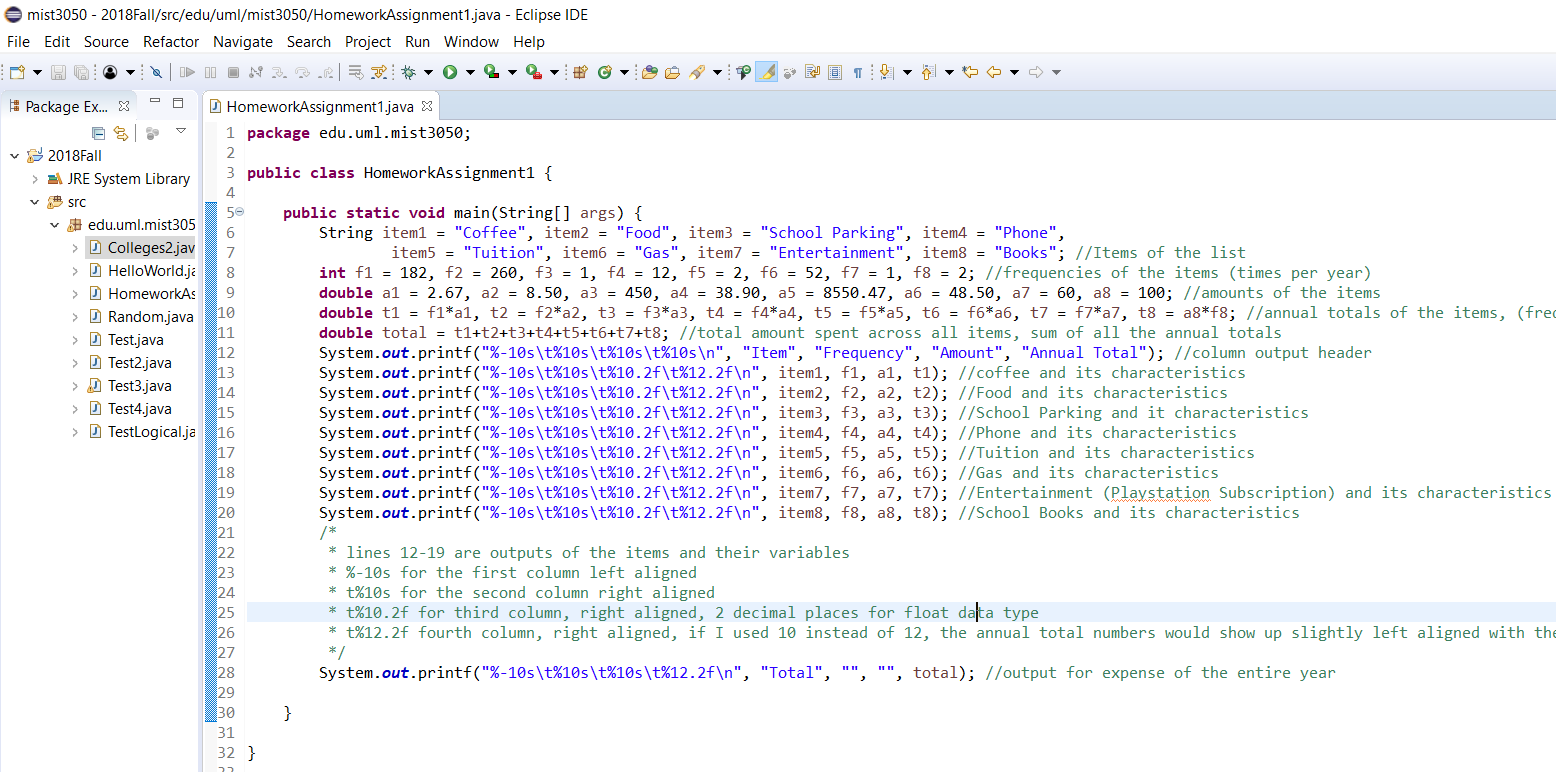
**Design and Implementation**

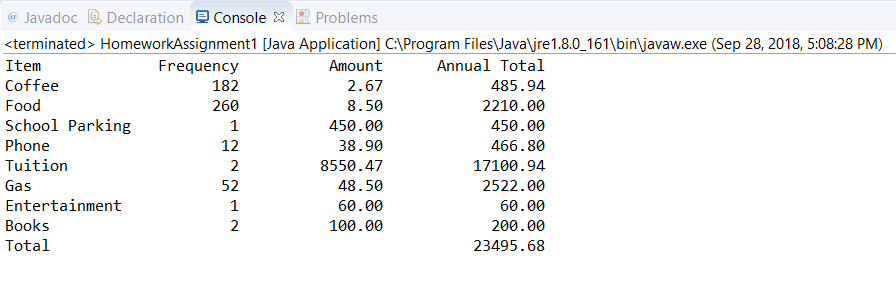
The structure of the program itself is straightforward, with variables and mathematical operations outlined in the beginning of the code, and the strings of output and their functions outlined in the middle and the end of the source code. There is only 1 class and method declaration for this code, though we are sure there are many ways to get to the same output instead of using one class and method. We didn’t think that this assignment had any concepts that were too complex, which is why we went for a more simplified approach at the program.

For the variables, we used a simple format by using a number and letter code for each variable. We used “item1, 2, 3… etc.” instead of “i1” because that would confuse us into thinking we were using imaginary numbers for our variables. The amounts were defined with “a”, their frequencies with “f”, their annual totals as “t” with a mathematical operator by multiplying the frequencies with their amounts, as well as a “total” variable for the grand total, a summation of the previous 8 annual total amounts.

The printf() function was used a lot throughout this assignment. We used different strings for different parts of the code, such as the column headers, the data, and the grand total. We enjoyed tweaking the numbers, defining the data types, and aligning the Strings part of the printf() to get the proper spacing, decimal places, and data types for our output.

**Testing**

 Here is the source code as well as its output:



The items are given the string data type, the frequencies are given the integer data type though if this record used weeks instead of days to keep track, then we could have used a double data type to show that. The amounts, annual totals and grand total are all given the double data type. What we noticed is that in lines 13 to 20, we used float “f” instead of double “d” to get the output to show proper decimals to the 2nd place, even though we defined those variables as double instead of float in the beginning of the code. There was no runtime error for this, so we assume that defining a variable and defining how it is shown in an output are different parts of the same code, because we did get a runtime error when we used “%10d” or “%10.2d”.

**Discussion**

We were able to get through this assignment with little to no difficulties. Perhaps the biggest difficulty was noticing that we forgot to place a { } or ; or “” somewhere and that would mess up the entire program. As mentioned before, there were some holes in our code that we thought would be detrimental to our execution of the program but were relieved when we found out that they weren’t.

The only source of help that we used was the Colleges java program that was used in Quiz 2 of the course. It helped us to define our strings and use the printf() function, which we used extensively. There is some Java experience within our group, as one of us has previously taken a similar course back in high school. It was easy for us to find our mistakes and correct them if there were any.

We learned that when it comes to programming, specificity is so important. Capital letters, proper punctuation, mathematical operators, and many more elements are crucial to the proper running of a program. It was fun for us to see Java again in a classroom setting, as it can be tough to teach the self how to code.

In terms of personal finances, this program is like keeping a personal checkbook or a company hiring someone to for their finances. It helps to record all transactions for tax and historical purposes. The items we used in the program are relative to each of us, and tuition is the catalyst to why our total expense exceeds $23,000. Obviously, this tuition will be paid back over time in the future, but we wanted to use as many items in this list that we could think of firsthand.

**Source Code**

**package** edu.uml.mist3050;

**public** **class** HomeworkAssignment1 {

**public** **static** **void** main(String[] args) {

String item1 = "Coffee", item2 = "Food", item3 = "School Parking", item4 = "Phone", item5 = "Tuition", item6 = "Gas", item7 = "Entertainment", item8 = "Books"; //Items of the list

**int** f1 = 182, f2 = 260, f3 = 1, f4 = 12, f5 = 2, f6 = 52, f7 = 1, f8 = 2; //frequencies of the items (times per year)

**double** a1 = 2.67, a2 = 8.50, a3 = 450, a4 = 38.90, a5 = 8550.47, a6 = 48.50, a7 = 60, a8 = 100; //amounts of the items

**double** t1 = f1\*a1, t2 = f2\*a2, t3 = f3\*a3, t4 = f4\*a4, t5 = f5\*a5, t6 = f6\*a6, t7 = f7\*a7, t8 = a8\*f8; //annual totals of the items, (frequency \* amount)

**double** total = t1+t2+t3+t4+t5+t6+t7+t8; //total amount spent across all items, sum of all the annual totals

System.***out***.printf("%-10s\t%10s\t%10s\t%10s\n", "Item", "Frequency", "Amount", "Annual Total"); //column output header

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item1, f1, a1, t1); //coffee and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item2, f2, a2, t2); //Food and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item3, f3, a3, t3); //School Parking and it characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item4, f4, a4, t4); //Phone and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item5, f5, a5, t5); //Tuition and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item6, f6, a6, t6); //Gas and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item7, f7, a7, t7); //Entertainment (Playstation Subscription) and its characteristics

System.***out***.printf("%-10s\t%10s\t%10.2f\t%12.2f\n", item8, f8, a8, t8); //School Books and its characteristics

/\*

\* lines 12-19 are outputs of the items and their variables

\* %-10s for the first column left aligned

\* t%10s for the second column right aligned

\* t%10.2f for third column, right aligned, 2 decimal places for float data type

\* t%12.2f fourth column, right aligned, if I used 10 instead of 12, the annual

\* total numbers would show up slightly left aligned with the column header

\*/

System.***out***.printf("%-10s\t%10s\t%10s\t%12.2f\n", "Total", "", "", total);

//output for expense of the entire year

}

}