**Title Page: cs 1011-051 – Lab 2 - Small Programs**

Cody Mott

9/20/2022

**I. Objectives**

This program is a culmination of small programs. The purpose of each is to get an input from the user, do something to that input, and print the result to the console.

negate gets an integer from the user, negates it, then prints it to the system output.

pennies gets a dollar amount from the user, multiplies it by 100, converts it to an integer, then prints it to the system output.

lastHalf gets a string of characters from the user, halves it, then prints the last half of the string to the system output.

makeInitialization asks the user for a variable name and an integer, it then prints that variable name an integer in an initialization statement to the system output, using the following format.

int *variableName* = *5*;

fraction asks the user for a numerator and a denominator, performs division on the two input integers, then prints the result to the system output.

**II. Requirements**

Write a class that contains the following 6 methods: main(), negate(), pennies(), lastHalf(), makeInitialization(), and fraction().

main() must call the other five methods in the order shown in the previous statement.

negate() must get an integer from the user, negate it using the unary operator **-**, then print the result to the console

pennies() must get a dollar amount from the user, convert it to cents and print the result to the console. input must be floating-point decimal and output must be an integer.

lastHalf() must get a word from the user and print the last half of the word to the console

makeInitialization() must get a variable name and an integer value from the user, the print them in an initialization statement to the console.

fraction() must get a numerator and a denominator from the user, perform division, and print the result to the console. the result must be a floating-point decimal.

Each method must display a title and ask nicely for inputs.

**III. Design:**

SmallPrograms Psuedocode:

START SmallPrograms

call main

call negate  
call pennies  
call lastHalf  
call makeInitialization  
call fraction

end main

END SmallPrograms

negate Psuedocode

START negate

print title  
declare input variable  
ask user for integer  
get input  
negate input  
print result

END negate

pennies Psuedocode

START pennies

print title  
 declare input variable  
 ask user for dollar amount  
 get input  
 convert to cents  
 print result

END pennies

lastHalf Psuedocode

START lastHalf

print title  
 declare input variable  
 ask user to enter a word  
 get input  
 print last half of word to console

END lastHalf

makeInitialization Psuedocode

START makeInitialization

print title  
 declare string variable  
 ask user for a variable name  
 get input string  
 declare integer variable  
 ask user for integer variable  
 get input integer  
 print initialization statement to console

END makeInitialization

fraction Psuedocode

START fraction

print title  
 declare numerator variable  
 ask user to enter a numerator  
 get input integer  
 declare denominator variable   
 ask user to enter a denominator  
 get input integer  
 perform division and initialize quotient variable  
 print quotient to console

END fraction

**III. Test Plan/Test Strategy**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Method name** | **Does title Print?** | **Does it ask for an input?** | **Correct type of input?** | **Correct output result?** | **Print result?** | **Work by itself?** | **Work in sequence?** | **PASS**  **/ FAIL** |
| negate |  |  |  |  |  |  |  |  |
| pennies |  |  |  |  |  |  |  |  |
| lastHalf |  |  |  |  |  |  |  |  |
| makeInitialization |  |  |  |  |  |  |  |  |
| fraction |  |  |  |  |  |  |  |  |

**IV. Results**

In the end, each method passed all the tests.

Graphical user interface, text

Description automatically generated  
Graphical user interface, text

Description automatically generated  
Graphical user interface, text

Description automatically generated  
Graphical user interface, application

Description automatically generated  
Graphical user interface, text, application

Description automatically generated

**V. Discussion**

I learned that the Scanner method nextLine() does not work well with other scanner methods. When nextLine() follows other scanner methods, it picks up the remaining whitespace from the previous input and does not look for the next input. To fix this issue, I just used the Scanner method next(). However, this limited me to just a single word input in my lastHalf() method. This still meets the requirements, I just wanted to have the ability to do an entire phrase.