CS 1050, Summer, 2021 Project #3 – Loops 100 points

Overview : Two separate programs using loops, Quizzes and Groceries. Work in teams.

Major topics

* while and for loops
* using sentinels for input
* testing

Program #1: Average Quiz Grades

Write a complete program that uses a for loop to read from the keyboard until up to 9 quiz grades are entered or -1 is entered before 9 grades are entered. The average of the grades, excluding the -1 in the count, is then calculated, a letter grade is assigned, and a report of both outputs is printed to the screen. Assume the user will enter at least one grade >= 0.

The report starts by asking for the user’s name. If the user presses <Enter> in response to the prompt, exit the program. As shown below, print a heading with the user’s name, then input and list the numbered quiz grades, then display the average quiz grade and the letter grade. Show the average quiz grade with one decimal place using the System.out.printf method (see Unit 1.6 in CS Awesome).

The letter grades are as follows: A: 90% to 100%, B: 80% to < 90%, C: 70% to < 80%, D: 60% to < 70%, F: < 60%. What form of the *if* statement works best in this situation?

Here is a sample report using five grades. **Match the format shown exactly** (including the quiz grades and blank lines):

Quiz Report for Susie

Quiz #1: 90

Quiz #2: 84

Quiz #3: 88

Quiz #4: 94

Quiz #5: 92

Quiz average: 89.6%

Quiz grade: B

You may assume all user input is correct in that grades between 0 and 100, or a -1 is entered. Use a *for* loop to input the quiz grades with a way to exit the loop early if the user enters a -1. Program logic:

Print an introduction to the program

Input the user's name

Exit if the user presses just the <Enter> key

Print the report heading

Set a grade counter to 0

Set a running total to 0

Loop up to 9 times

prompt for a quiz grade

read the quiz grade

if the user enters -1, exit the loop

increment the grade counter

print the quiz grade

add the quiz grade to the running total

Calculate the average grade from the running total and grade counter

Print the average grade, formatted

Determine and print the letter grade

--------------------------------------------------------------------------------------------------------------------------------------------------

Program #2: Grocery Bill

Write a complete program that reads and prints a grocery bill. Detailed program logic is not supplied.

Using a *while* loop, prompt the user for grocery item data. For each item, input the item name (e.g., York Peppermint Patties), the price, and whether the item is on sale (y/n, in either upper or lower case). If the user presses just the <Enter> key for the item name, exit the loop, print the results and exit the program. You may assume all user input is correct. All dollar amounts are rounded to two decimal places.

Print the grocery bill listing as shown below. For items on sale, print the word SALE followed by the sale price. Items on sale get a discount of 20%.

Finish the bill with the total price for all the items, a tax amount of 7%, and the final bill (total of items plus tax).

Partial sample report, match the format shown.

Grocery Bill

Purchases

cereal: $2.50

soda pop: $2.00 SALE: $1.60

: << more items

:

<< blank line before the total

Subtotal: $27.50 Line the 3 dollar amounts with the decimal places; use a 7% tax rate

7% tax: $1.92

Total bill: $29.42

--------------------------------------------------------------------------------------------------------------------------------------------------

Testing

Test data is posted under Modules > Projects for each program (multiple sets of quiz grades and multiple sets of grocery bills). The names of the test files are:

* 1050 - Project 03 - Test Data - Quiz.txt
* 1050 - Project 03 - Test Data - Grocery.txt

Enter the data manually for each test case. Submit the data separately from the test cases for Quiz and Grocery as your output files. You will have to copy and paste the results of each run into the respective output file. Run all the test cases and check the results for accuracy, including formatting.

You should test as you develop the programs, not when they are all finished. For example, you can enter one grocery item's name, without a price, and see if the beginning of the bill prints out correctly (heading and item name). Then add the price and check again. Then make it a sale item and check the discount. When one item works right, add the loop and test sets of items.

Create test results documents with two reports: the output from the quizzes test cases and the output from the groceries test cases.

Submission of your project means that all given test data runs correctly.

Documentation

Follow the style guidelines in Foundation > file # 56 Java Programming Style Guide.

--------------------------------------------------------------------------------------------------------------------------------------------------

Submission & Grading

Target completion: Monday, July 12

Due date: Thursday, July 15

On-time projects must compile without error, run without crashing, and get correct results. Projects will be accepted up to one week late, with a point loss of 20%.

Submit one set of files for each team, not one set for each team member. Use one team member’s name for the file names. No PAGES files! Your team’s submission should have these files (no zip files) in this order:

* FirstnameLastname\_03\_Cover.docx Cover letter (use the template; see Modules > Projects)
  + THE FIRST LINE OF THE COVER LETTER MUST HAVE YOUR TEAM MEMBER NAMES
* FirstnameLastname\_03\_Quiz.java
* FirstnameLastname\_03\_Quiz\_Output.txt (or pdf or docx or …,)
* FirstnameLastname\_03\_Grocery.java
* FirstnameLastname\_03\_Grocery\_Output.txt (ditto)

Do not submit the test files as they are the same for everyone.

--------------------------------------------------------------------------------------------------------------------------------------------------

Notes

* Ask if you are uncertain about any specifications.
* Go to Warren’s office hours with any problems or email/call me. Remember the 30-minute guideline to ask for help.
* You are welcome to submit your completed project ONCE for a broad review before actual submission.
* All specifications must be followed, including output format.

--------------------------------------------------------------------------------------------------------------------------------------------------

Challenges (optional, no extra credit)

* Quizzes: add a feedback comment to the student after each letter grade. For example, the report might say "Great job!" if the students earns an A. The comment should be on the same line as the letter grade.
* Groceries: count and print the total number of items the user purchased.
* Groceries: calculate and print the total of how much the customer saved on the sale items.

--------------------------------------------------------------------------------------------------------------------------------------------------