

Programming Assignment 3/4

Scenario:

Working at Mason Dining, you are asked to build a small application to help track the dining bucks associated to a specific student. All students start with \$106 in their account. They may add funds or eat a meal, according to the following prices:

- Each breakfast costs \$4.92
- Each lunch costs \$7.06
- Each dinner costs \$10.60

Create a program that will continuously prompt the user for one of three actions: adding funds to their account, eating a meal, or exiting the program. When the user wishes to add funds, they must be prompted to put in a dollar amount to add to their account. When the user wishes to eat a meal, they must select which meal they are eating. Users without sufficient funds must be told they have insufficient funds. In both cases, the program should maintain an accurate account balance.

After the user has indicated they wish to exit the program (this is the only way the program should exit), the program will display a well-formatted report containing the starting account balance, total number of meals eaten, and final account balance.

You must check that each type of user input is valid. (*Hint: Remember Double.parseDouble(), using if statements, and try/catch. Review the readings and notes for examples.*). When input is invalid, the user must be informed of an error and re-prompted.

To Do (Check Blackboard for Due Dates):

Programming Assignment 3: Solution Design

- 1) Create a defining diagram that shows the input, processing, and output
- 2) Create a solution algorithm using pseudocode
- 3) Show testing using the desk checking table method, to include test data, expected results, and a desk checking table. Make sure your desk checking considers multiple cases including both valid and invalid test data to prove your algorithm will work

Upload a Word document containing only items above to Blackboard.

Grading Criteria	
Requirement	Points
Defining Diagram with input, processing, and output	40
Efficient Solution Algorithm	40
Thorough Desk Checking Table including test data, and expected results	20

Full points will be awarded for an accurate, efficient, complete defining diagram, solution algorithm, and desk checking table. Partial credit is available.

Programming Assignment 4: Solution Implementation

Write a well-documented, efficient Java program that implements the algorithm you identified. Include appropriate documentation as identified in the documentation expectations document.

Note: You may not use the Scanner or System.out classes. For input/output, you must use the JOptionPane class.

Upload the .java file of the final program to Blackboard.

Full points will be awarded for an accurate, efficient, complete Java program. Partial credit is available. Any final program that does not compile will receive an automatic zero.

Grading Criteria	
Requirement	Points
Implementation of Java Program, using efficient practices where appropriate, such as the use of constants, good variable names, no redundant code, etc.	70
Appropriate objective-style documentation	10
Appropriate intermediate comments	20