**Projects for Chapter 08**

## Project 8-1: Tip Calculator

Add exception handling to a Tip Calculator program.

## Console

Tip Calculator

INPUT

Cost of meal: ten

Must be valid decimal number. Please try again.

Cost of meal: -10

Must be greater than 0. Please try again.

Cost of meal: 52.31

Tip percent: 17.5

Must be valid integer. Please try again.

Tip percent: 20

OUTPUT

Cost of meal: 52.31

Tip percent: 20

Tip amount: 10.46

Total amount: 62.77

## Specifications

* The program should accept decimal entries like 52.31 and 15.5 for the cost of the meal.
* The program should accept integer entries like 15, 20, 25 for the tip percent.
* The program should validate both user entries. That way, the user can’t crash the program by entering invalid data.
* The program should only accept numbers that are greater than 0.
* The program should round results to a maximum of two decimal places.

## Project 8-2: Wizard Inventory

Add exception handling to a program that keeps track of the inventory of items that a wizard can carry.

## Console if the program can’t find the inventory file

The Wizard Inventory program

COMMAND MENU

walk - Walk down the path

show - Show all items

drop - Drop an item

exit - Exit program

Could not find inventory file!

Wizard is starting with no inventory.

Command: walk

While walking down a path, you see an unknown potion.

Do you want to grab it? (y/n): y

You picked up an unknown potion.

Command: show

1. an unknown potion

Command: drop

Number: x

Invalid item number.

Number: 1

You dropped an unknown potion.

Command:

## The error message if the program can’t find the items file

Could not find items file.

Exiting program. Bye!

## Specifications

* Store a list of all possible items that a wizard can carry in a file. When the user selects the walk command, read these items from the file, randomly pick one, and give the user the option to grab it.
* Store the current items that the wizard is carrying in another file. Make sure to update this file every time the user grabs or drops an item.
* The wizard can only carry four items at a time.
* For the drop command, display an error message if the user enters an invalid integer or an integer that doesn’t correspond with an item.
* Handle all exceptions that might occur so that the user can’t cause the program to crash.
* If the items file is missing, display an appropriate error message and exit the program.
* If the inventory files is missing, display an appropriate error message and continue with an empty inventory for the user. That way, the program will write a new inventory file when the user adds items to the inventory.

## Project 8-3: Monthly Sales

Add exception handling to a program that reads the sales for 12 months from a file and calculates the total yearly sales as well as the average monthly sales.

## Console

Monthly Sales program

COMMAND MENU

monthly - View monthly sales

yearly - View yearly sumary

edit - Edit sales for a month

exit - Exit program

Command: edit

Three-letter Month: Dec

Sales Amount: TK

Sales amount for Dec was modified.

Command: monthly

Jan - 14317

Feb - 3903

Mar - 1073

Apr - 3463

May - 2429

Jun - 4324

Jul - 9762

Aug - 15578

Sep - 2437

Oct - 6735

Nov - 88

Dec - TK

Command: yearly

Using sales amount of 0 for Dec.

Yearly total: 64109

Monthly average: 5342.42

Command: exit

Bye!

## Specifications

* Use a CSV file to store the data for the program.
* If the program can’t find the CSV file when it starts, display an error message and exit the program.
* For the edit command, display an error message if the user doesn’t enter a valid three-letter abbreviation for the month.
* When the user edits the sales amount for a month, the data should be saved to the CSV file immediately. That way, no data is lost, even if the program crashes later.
* If the CSV file doesn’t contain a valid integer for the sales amount for the month, use a value of 0 to calculate the total sales for the year.
* Round the results of the monthly average to a maximum of 2 decimal digits.