

CS 1063 Fall 2022 Project 1: Grade System

(50 points)

Objectives

This is one of two major programming projects this semester. You should NOT collaborate your code/script on this project. While you may ask for assistance in debugging, this project should be ENTIRELY your own work.

Objectives include:

- Use variables
- Print information to and pull information from the console
- Use Sequence Types
- Use If Statements
- Use loops
- Use functions

Hand-in Requirements

All projects will be submitted electronically through Blackboard. Zip up your entire project directory/folder to submit. (Right-click on the project folder and select zip or compressed files) Use the following naming convention for your submission: << **Project1_abc123.zip** >> The project folder should include all files necessary to run your program. It must run in Spider version 5.1.5 (the Spider version on the UTSA VDI).

Introduction

This project aims to help a UTSA instructor manage and generate reports of students' grades. Your program will allow the teachers to:

- Create a dictionary of students and their exam grades by inputting information in through the console.
- Based on a menu of choices:
 - List all student names with exam grades
 - Create a report based on letter grades
 - Modify an existing grade for a student
 - Exit the Program

You will be asking your instructor/user a number of questions. The test script used to evaluate your program is provided at the end of this document. In addition, a rubric is provided to help you understand the requirements.

Details

The program has two sections

1. Create and populate a dictionary type variable with student names and grades
2. Present a menu to the instructor/user to manage information in the dictionary

Section 1

You must include a header. The header must follow the format below. (Insert your full name in the YOURNAME).

UTSA – Fall 2022 – CS1063 – Section 001 – Project 1 – written by YOURNAME

You must include a prompt to the instructor/user to enter the class size. For this program, we will not exceed 10 students.

Please, enter the class size:

For each member of the class, you will prompt the instructor/user for the student's name and grade. These values must be included in your dictionary type variable.

Please, enter the student name:

Please, enter the student grade:

Once all students and their grades are entered, the program will advance to section 2.

Section 2

The program will present a menu to the instructor/user, with 4 options.

Main Menu

1 – List Student Grades, 2 – Report of Grades, 3 – Modify a Grade, 4 – Exit Program

Selection an Option:

Based on console input from the instructor/user, the program will perform one of the four below activities. It will continue to prompt the instructor/user until option 4 (Exit Program) is selected. Then the program will close with a farewell message.

- List all student names with exam grades
- Create a report based on letter grades
- Modify an existing grade for a student
- Exit the Program

Note: If the instructor/user inputs a number outside of 1-4, the program should prompt the user to resubmit due to an error. The error message is below.

Value is out of range, please resubmit:

1 - List Option

When the instructor/user selects option 1, the program will display all students with their associated grades in the below format. The display will start with a header and each student will be listed on a separate line and the #s represents their grade.

List of Student Grades:

name's exam grade is ###.#

name's exam grade is ###.#

...

2 - Report of Grades

When the instructor/user selects option 2, the program will generate a report that reports the number of grades per letter grade. Letter grades are based on the ranges listed in the table below.

Letter	Range
A	90 <= grade <= 100
B	80 <= grade < 90
C	70 <= grade < 80
D	60 <= grade < 70
F	grade < 60

The display will start with a header and list the total number of grades for each letter grade.

Report of Grades

A : ##

B: ##

C: ##

D: ##

F: ##

3 – Modify a Grade

When the instructor/user submits option 3, the program will prompt the instructor/user to provide the name of the student.

Please provide the name of the student:

Note: if the instructor/user does not provide a valid name (key), the the program should prompt them to resubmit.

Invalid Name, please resubmit:

After the program gets a valid student name, it will prompt the instructor/user to provide the new grade.

Please provide the new exam grade:

After the grade is input, the program will print the student's name and new grade

Studentname's new exam grade is ##.##:

4 – Exit Program

When the instructor/user submits option 4, the program will display the below farewell message on the console. Include your name in YOURNAME>

Thank you. This program is brought to you by Python Expert: YOURNAME.

Test Script

The below values are the ones that will be used to test your program. We will be inserting the values in the order below. Each value will be input one at a time.

Section 1 Test Values:

5, Linda, 100, Tom, 82, Steve, 50, Cindy, 83, Jim, 77

Section 2 Test Values

7, 1, 2, 3, Steve, 90, 1, 2, 3, Jim, 84, 1, 2, 3, Robert, 4

Rubric

Your program should compile with no errors. A program with more than one or two errors will likely get a zero for the whole assignment. The following criteria will also be used to determine the grade for this assignment:

- [5 points] If your submission includes the following:
 - The program displays in the header and footer with your specifics (name).
 - Your submission was a Zip file named per instructions.
 - Your program contains comments that describe what the program does. Your program includes a comment describing each function, essential variables, and significant portions of the program.
 - Your program is appropriately indented.
- [5 Points] The first section of your program builds the grades dictionary variable.
 - You use the prompts as described
 - The dictionary variable includes all the required keys and values
 - You can use the dictionary variable across the entire program
- [5 Points] The second section part of the program successfully processes the menu options
 - You use loops to successfully rotate through the options
 - Implementing the loop until the user selects choice 4
 - Logic for the right options applied to call the different functions
 - Correct error message if user does not put in valid selection value
 - [-20 Points] If you do not follow the menu as described in instructions
- [10 Points] The list option was implemented following the described process.
 - Use function to perform list option
 - Display matches the instructions
- [10 Points] The Report Grades option was implemented following the described process.
 - Use function to perform Report Grades option
 - Correct logic is used to process the letter grades
 - Display matches the instructions
- [10 Points] The modify Grade option was implemented following the described process.
 - Correct error checking is performed per instruction
 - Correct modification is made to dictionary values
 - Display matches the instructions
- [5 Points] The exit option was implemented following the described process