

CS 1400 – Summer 2018

Assignment #9

Introduction

You will complete one program to give you experience with:

- Classes
- Private data
- Getters & Setters
- Constructors
- Access Operator

Task 1

See the demo video for this assignment. Use the provided code file to complete the task to match the video. You must fill in the blanks (`<Fill-In>`), as well as add any other methods or data that is needed. All blanks should be completed with a single line of code (or partial line). You should not modify any other code that exists. Only remove the blanks and add other missing methods.

Use a radius of 100 for the head. The mouth is a thicker line. The face is centered at 0, 0.

Rubric

0 pts: Software Development Lifecycle Plan not necessary this time

10 pts: main() blanks filled in properly with single lines/partial lines

10 pts: Face class blanks filled in with single lines

10 pts: Appropriate Face class methods added

10 pts: Initial face draws properly

10 pts: Face is modified properly

Task 2

Programming Exercise 7.3

Read the exercise in the book first, so these instructions make sense. Instead of writing the test program as instructed in the book, write a program that obtains user input to use the Account class. The program should ask for the starting values for id, balance, and interest rate. The program then should present a menu where a user can access/modify their account. After each selection an appropriate message should be displayed. The menu should then be displayed again (Unless 'Exit' is selected)

Example Menu:

- (1): Display ID
- (2): Display Balance
- (3): Display Annual Interest Rate
- (4): Display Monthly Interest Rate
- (5): Display Monthly Interest
- (6): Withdraw Money
- (7): Deposit Money
- (8): Exit

Make sure to read the rubric to see the additional requirements. Note: you do not need to write the program to test as described in the book. You will be writing a different program to use the class.

Rubric

5 pts: Software Development Lifecycle Plan (see assn #5 for description)

5 pts: Accurate UML diagram

- 5 pts: Implementation meets all requirements listed in book**
- 5 pts: Account class is in its own module file account.py**
- 5 pts: Follow proper coding conventions**
- 5 pts: Input obtained properly from user**
- 5 pts: Controlling input**
 - You may assume all input received are proper numbers**
 - You should not allow any negative numbers**
 - Interest rate should not be greater than 10%**
 - Loop and request input again if an invalid number is entered**
- 10 pts: All calculations are accurate**
- 5 pts: Menu loops properly until user enters '8'**
 - You may assume that all input are valid integers**
 - Print a message and loop back to menu if an invalid menu number is entered**

Starter

Use the main.py file to get started. Only replace the blanks and add missing methods. Do not modify the existing code otherwise

Helpers

Remember that you can find solutions to the even programming exercises online. Check Canvas for a link. These are suggestions for you to do. They are not part of the assignment, and you do not have to turn them in.

Exercises: 7.2, 7.4, 7.10

What/How To Turn In (READ THIS)

Submit your files on Canvas.

Due: June 9, 2018