CSC 242: Introduction to Computer Science II

Assignment 4

Due Tuesday, April 28th, at 11:59pm

Reading

Review Chapter 8 and read Chapter 9 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

Logistics

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in a comment at the top of the assignment, and the number of collaborators on any assignment **may not exceed two other people**. You must also submit a comment in your submission for each assignment that describes in detail how each collaborator contributed to the assignment. If you did not collaborate with anyone on the assignment, you must include a comment that says that. You may not under any circumstances discuss the assignments with classmates' other than your identified collaborators. Working so closely with anyone other than your identified collaborators, Mr. Zoko, or the lab assistant, so as to produce identical or near identical code is a violation of the Academic Integrity policy. This policy will be strictly enforced.

Please include the following with your assignment submission:

1. A comment at the top of your Python file identifying any classmates with whom you discussed or in any other way collaborated on the assignment. You may work (directly or indirectly) with **no more than two** other people.
2. Add a comment at the top of your Python file that describes for each person what they contributed to the assignment. This must be at least 2-3 sentences and be **very specific and detailed**.

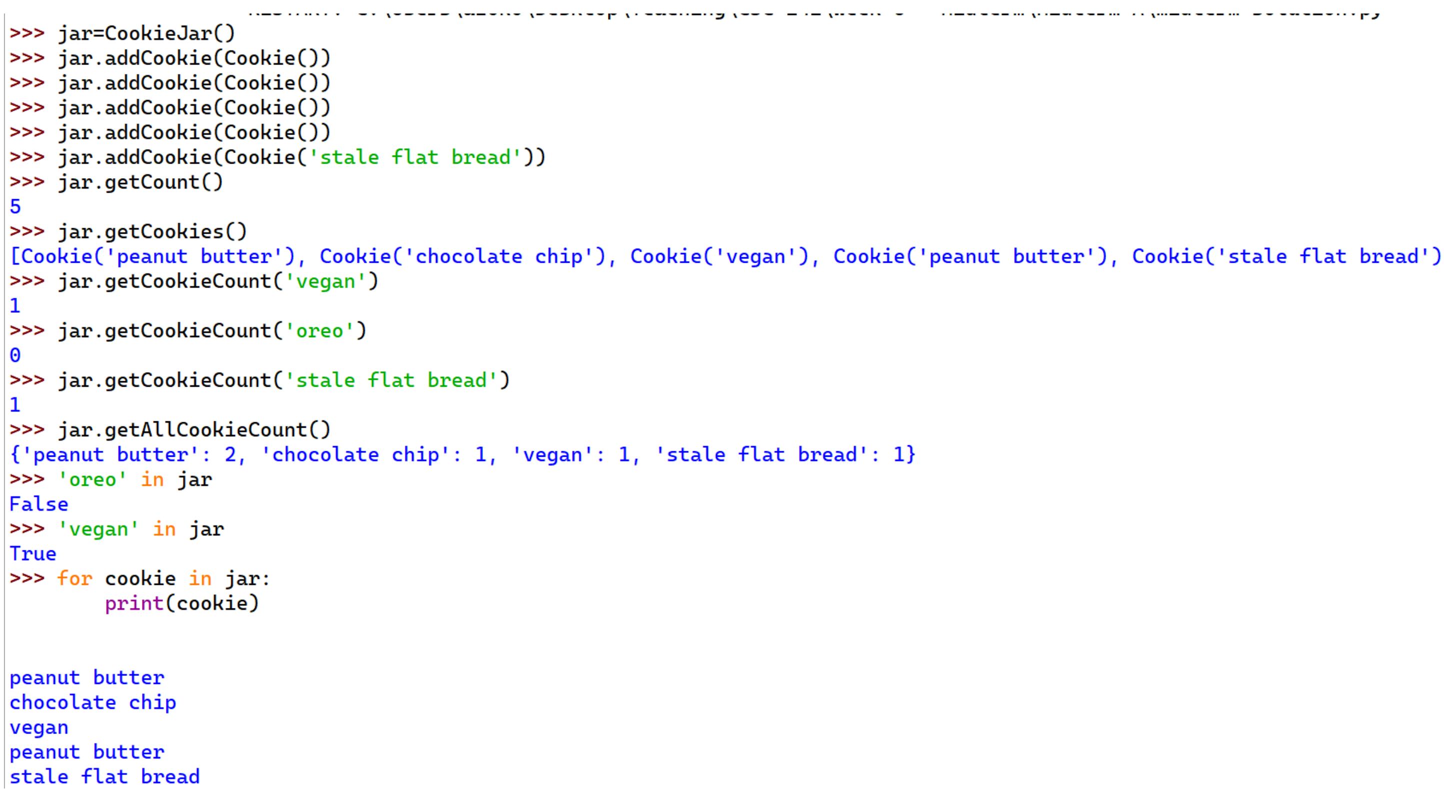
A submission that does not include a list of collaborators and a comment indicating how you collaborated with classmates will earn a 0. If you worked alone you must put a comment at the top of your file that indicates that or you will also receive a 0. There will be no exceptions to this rule.

Again, you are subject to all the rules specified in the Academic Integrity policy. Please read it carefully before beginning this assignment.

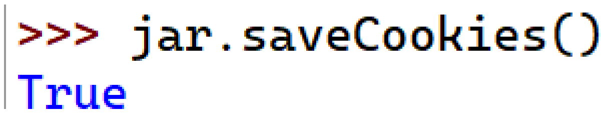
Assignment

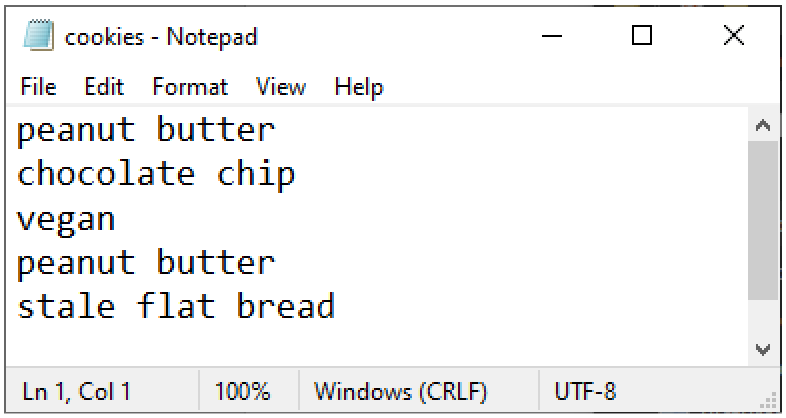
Begin the assignment by downloading the **csc242hw4.py** file from the D2L site.

Problem 1 (30 points) : Implement the CookieJar object as shown in the screenshots. ***You should not change the Cookie implementation***. Since the Cookie object picks a random type of cookie if a cookie type is not specified, your results may be slightly different than mine below. The docstrings in the template contain descriptions of each method.



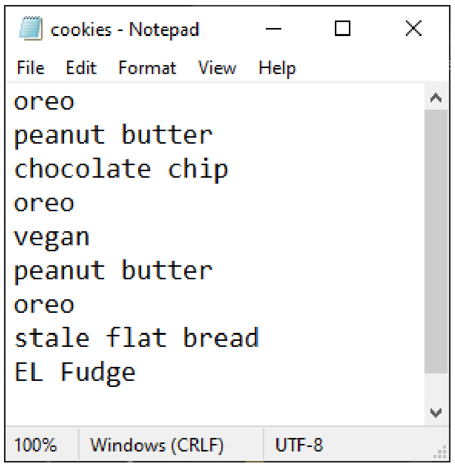
```
>>> jar=CookieJar()
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie('stale flat bread'))
>>> jar.getCount()
5
>>> jar.getCookies()
[Cookie('peanut butter'), Cookie('chocolate chip'), Cookie('vegan'), Cookie('peanut butter'), Cookie('stale flat bread')
>>> jar.getCookieCount('vegan')
1
>>> jar.getCookieCount('oreo')
0
>>> jar.getCookieCount('stale flat bread')
1
>>> jar.getAllCookieCount()
{'peanut butter': 2, 'chocolate chip': 1, 'vegan': 1, 'stale flat bread': 1}
>>> 'oreo' in jar
False
>>> 'vegan' in jar
True
>>> for cookie in jar:
        print(cookie)


peanut butter
chocolate chip
vegan
peanut butter
stale flat bread
```


```
>>> jar.saveCookies()
True
```



```
peanut butter
chocolate chip
vegan
peanut butter
stale flat bread
```

If I change the contents on cookie.txt to the following and load a new jar:



```
oreo
peanut butter
chocolate chip
oreo
vegan
peanut butter
oreo
stale flat bread
EL Fudge
```

```
>>> jar=CookieJar()
>>> jar.loadCookies()
True
>>> jar.getAllCookieCount()
{'oreo': 3, 'peanut butter': 2, 'chocolate chip': 1, 'vegan': 1, 'stale flat bread': 1, 'EL Fudge': 1}
>>> for cookie in jar:
        print(cookie)


oreo
peanut butter
chocolate chip
oreo
vegan
peanut butter
oreo
stale flat bread
EL Fudge
```

Problem 2 (30 points) : You need to create a class named AverageCalculator that inherits from object. A template can be found on the next page and in the .py file.

Method Description:

a. __init__(self,fileName='grades.txt') – Takes in a parameter of a filename that has a list of grades for students. If none is specified, the filename defaults to 'grades.txt'

b. loadGrades(self): Load grades opens the file and reads in a list of lists of student grades. A line in the file represents the grades for one student. Returns True if successful and False if not.

c. writeLetterGrades(self): This method calculates the letter grade of the students and writes the letter grades to a file. ***You must call getStudentAverages() to get each students average in the course***. If the file already exists, it is overwritten. Returns True if successful and False if there is an error. The method then assigns a letter grade as follows:

    - Between 100 and 90 : A
    - 89 to 80 : B
    - 79 to 70 : C
    - 69 to 60 : D
    - Less than 60: F

   The letter grades are written to a file named avg- then followed by the name of the file you read from. For example, if the class was initialized with file grades.txt, the letter grades are written to file avg-grades.txt (avg- is concatenated to the filename). Each line of the file is the average of one students grades and the letter grade. Example output to file:

```
85.0:B
71.75:C
100.0:A
```

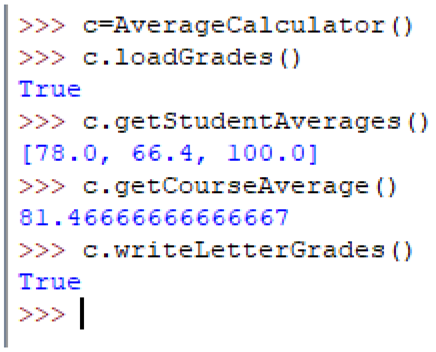
d. getStudentAverages(self): Takes the data read in by a call to loadGrades() and returns a list of averages of the student grades.

e. getCourseAverage(self): Returns the average of all the student grades.

Sample Usage:

Grades.txt file:

80,90,50,70,100
50,67,45,90,80
100,100,100,100,100



```
>>> c=AverageCalculator()
>>> c.loadGrades()
True
>>> c.getStudentAverages()
[78.0, 66.4, 100.0]
>>> c.getCourseAverage()
81.46666666666667
>>> c.writeLetterGrades()
True
>>>
```

Contents of avg-grades.txt:

78.0:C
66.4:D
100.0:A

Problem 3: 40 Points. Develop a GUI named **MortgageCalculator** that can be used to calculate the monthly payment of a mortgage.

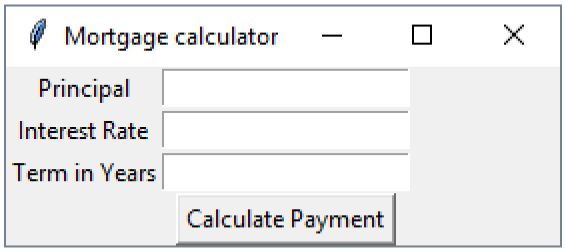
The mortgage monthly payment can be calculated as follows:

- P = Principal (the amount of the loan) as a decimal
- N = Term of Loan (how many years to pay off the loan) as an int.
- R = The interest rate on the loan as a decimal.
- The values will be used to compute the following formula:

$$m = \frac{p\ r\ /\ 1200.0}{1 - (1.0 + r\ /\ 1200.0)^{-12.0\ n}}$$

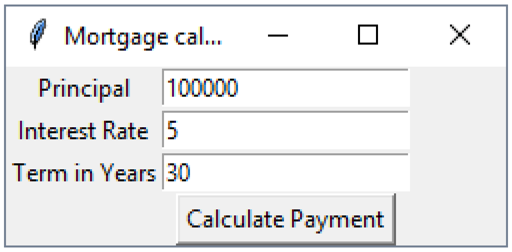
Test your calculation/output here: https://www.easycalculation.com/mortgage/loan-payment-amount.php

This is the default view. **You need to use grid() for layout**.



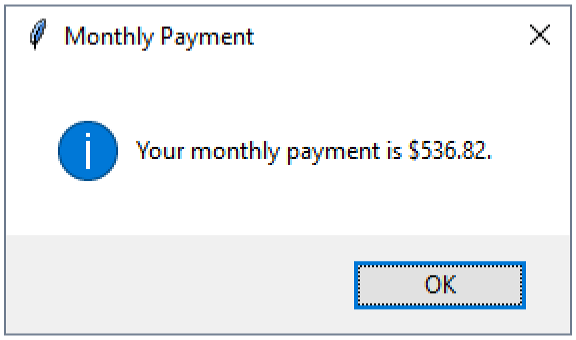
The layout consists of 3 labels, 3 entry text boxes and 1 button.

For example. When calculating the mortgage for the following values:



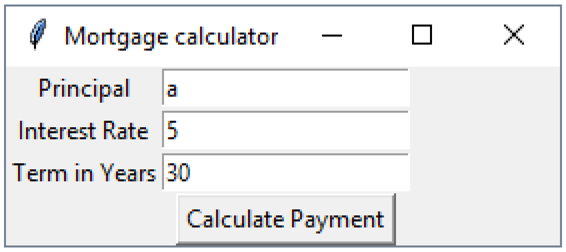
Principal 100000
Interest Rate 5
Term in Years 30

A popup shows the result. Notice formatting! I will be taking 10 points off for incorrect format! The values should be treated as float().



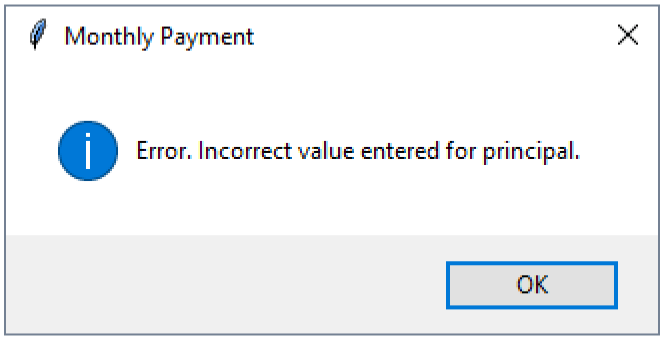
Your monthly payment is $536.82.

User enters an incorrect value:



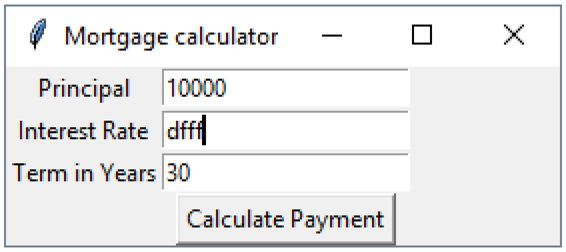
Principal a
Interest Rate 5
Term in Years 30

They are warned that an incorrect value was entered for principal.

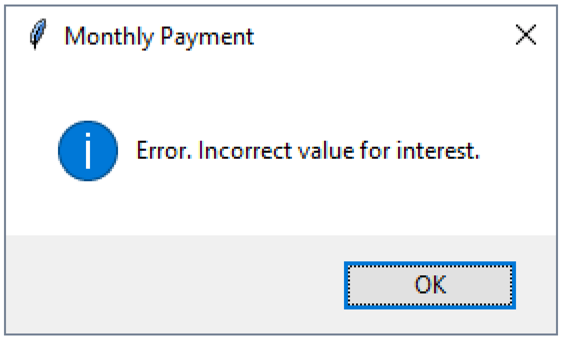


Error. Incorrect value entered for principal.

This applies to Interest and Terms as well:

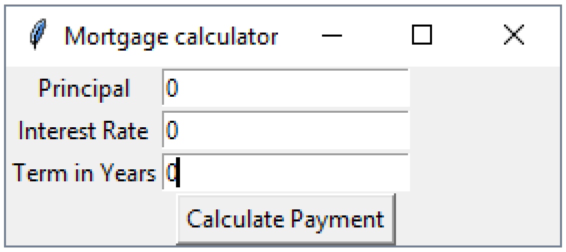


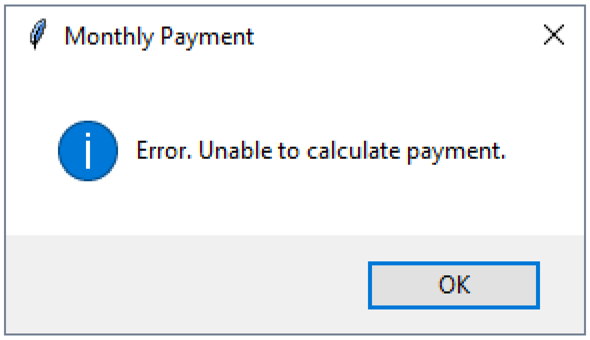
Principal 10000
Interest Rate dfff
Term in Years 30



Error. Incorrect value for interest.

What can happen if the users all 0s? Divide by Zero! **All errors should be caught**!




Monthly Payment

Error. Unable to calculate payment.

You will run the program using **MortgageCalculator().mainloop().**

Do not change methods provided in the template. Instead you must fill in all the methods.

Submitting the assignment

You must submit the assignment using the assignment 4 dropbox on the D2L site. Submit a Python file (csc242hw4.py) with your implementation in it and comments describing your collaboration status. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The assignment is worth 100 points. Any student who does not submit comments in the Python file describing the contributions of each team member or indicating that he/she worked alone will earn a 0 on the assignment.