

Honors Computer Science Python

Mr. Clausen

Program 5A, 5B, 5C, 5D

PROGRAM 5A I'm Sort of Searching for Monty Python (35 points)

This program is going to practice the sorts and searches that we cover in every programming language. Python has some built in methods to help achieve this goal. This program will practice the Python list methods as well as the standard algorithms.

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) Initialize all of the variables that are to be used in this program.
- 3) Use print statements to display your name and period output just like those used for program 1A.
- 4) Display the menu (as pictured below) and ask the user to enter their choice.
- 5) For this program we will implement menu choices 1, 2, 0 (zero), P, S, I, and D in that order. Upper and lower case values should work for the menu choices.
- 6) Make sure that you use descriptive identifiers for all of your variables including names for the lists. Please use the names **originalList** and **listCopy**. The word "list" is a reserved word in Python (menu choice #2).
- 7) In menu choice #1, ask the user how many elements they want in the list. Generate random integers from 1 to this value to fill the list with that many elements.
- 8) In menu choice #2, properly copy the list. For menu choices P, S, I, and D use Python's List Methods to implement these choices. For all the other menu choices use a 'print' statement that says that this menu choice hasn't been implemented yet.
- 9) For menu choice "S" ask the user for the number they wish to find. Display the index number of every occurrence of this number as well as how many times the number was found in the list.
- 10) Save your program as `LastNameFirstNameP5A.py`.

I'm Sort Of Searching For Monty Python

1. Generate Random Numbers For The Original List
 2. Copy The Original List
 3. Sequential (Linear) Search
 4. Binary Search (The List Must be Sorted)
 5. Selection Sort
 6. Bubble Sort
 7. Insertion Sort
 8. Merge Sort
 9. Quick Sort
 - P. Python's Sort Method
 - S. Python's Search and Count Method
 - I. Insert an element in the List at any position
 - D. Delete an element from the List at any position
 0. Display the numbers using a 'loop' (press zero)
- Press enter without choosing anything to quit

Enter your choice:

PROGRAM 5B Functional Sphere (35 points)

This program is going to practice writing functions as we rewrite Program 2A using functions. Write a program to calculate the diameter, circumference, surfaceArea, and volume of a sphere. Ask the user to enter a value for the radius. Make your program user friendly by prompting them for this value. Use one line comments to separate this program into its parts

- 1) Use a DocString at the beginning of the program for your comments.
- 2) Import math.
- 3) Initialize all of the variables that are to be used in this program **in the main function** (make sure that you use descriptive identifiers for all of your variables) and call all of the other functions from the main function.
- 4) Use print statements to display your name and period output just like those used for program 1A in a function named displayMyInfo.
- 5) Ask the user for the radius in a function named getData.
- 6) Perform all of the calculations in separate functions named calcDiameter, calcCircumference, calcSurfaceArea, and calcVolume.
- 7) Echo out the value of the radius as well as the results of your calculations for the diameter, circumference, surfaceArea, and volume of a sphere in a function named displayOutput.
- 8) Make sure you have DocStrings for each function including a brief description of the function, the type contract, and some sample function calls with sample expected outputs.
- 9) Save your program as LastNameFirstNameP5B.py.

```
diameter = 2 * radius
circumference = diameter * pi
surface area = 4 * pi * radius * radius
volume = 4/3 * pi * radius * radius * radius
```

Use the exponent calculation instead of multiplying the radius by itself.
Use the value of pi contained in the math module you imported.

PROGRAM 5C Compounding Your Interest (40 points)

This program is going to practice writing functions as we solve a compound interest problem. Write a program to calculate the final amount and interest earned or owed. Ask the user to enter the principal, interest rate, number of times the interest is compounded per year, and the number of years that the principal is invested or borrowed. Each of these inputs need their own separate function since we return only value from each function. Make your program user friendly by prompting them for these values. Use one line comments to separate this program into its parts

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) Initialize all of the variables that are to be used in this program **in the main function** (make sure that you use descriptive identifiers for all of your variables) and call all of the other functions from the main function. I know that this formula is a “famous formula”, but for this program don’t use any one letter variables. Please use the variable names: interest, principal, amount, time, rate, and numCompounded.
- 3) Use print statements to display your name and period output just like those used for program 5B in a function named `displayMyInfo`.
- 4) Ask the user for the principal in a function named `getPrincipal`.
- 5) Ask the user for the length of time for the investment or loan in a function named `getTime`.
- 6) Ask the user for the number of times the interest is calculated per year in a function named `getNumCompounded`.
- 7) Ask the user for the interest rate expressed as a decimal (ie. 5% = 0.05) in a function named `getInterestRate`.
- 8) Perform all of the calculations in separate functions named `calcAmount`, and `calcInterestOnly`.
- 9) Echo out all of the values for principal, time, number of times compounded, and interest rate in a function named `displayOutput` as well as displaying the final amount and interest earned/owed.
- 10) Make sure you have DocStrings for each function including a brief description of the function, the type contract, and some sample function calls with sample expected outputs.

11) Save your program as LastNameFirstNameP5C.py.

```
amount = principal * (1+ rate/numCompounded)**(numCompounded*time)
interest = amount – principal
```

PROGRAM 5D PhoneBook (50 points)

This program is going to practice writing functions as we also practice Python dictionaries. Write a phone book program to perform all of the operations illustrated in the menu below. Each of these menu choices need their own separate function. Make your program user friendly by prompting them for these values. Use one line comments to separate this program into its parts

```
Phone Book
1. Look up a phone number
2. Add a phone number
3. Change a phone number
4. Delete a person and their phone number
5. Display all names and phone numbers in the phone book
6. Display all the names in the phone book in alphabetical order
7. Display the dictionary and inverted dictionary for debugging purposes
8. Display all the names & numbers in the phone book in alphabetical order
9. Reverse phone number lookup
Q. Quit the program
Enter your choice
```

- 1) Use a DocString at the beginning of the program for your comments.
- 2) Initialize all of the variables that are to be used in this program **in the main function** (make sure that you use descriptive identifiers for all of your variables). Call the function displayMyInfo, and then call two other functions from the main function: the functions “menu” and “controlMenuExecution”. These last two functions should be inside a while loop that will continue until the user enters a “Q” or a “q” to quit the program. Pass the two dictionaries phoneBook and invertedPhoneBook to the controlMenuExecution function and from there to any other function that needs them. Remember to return any dictionary that is changed in a function back to the calling function. Use the names phoneBook and invertedPhoneBook for the dictionaries. Phone book should be initialized with two entries to start, while the inverted phone book dictionary should be empty at the beginning of the program: invertedPhoneBook = {}.
- 3) Use print statements to display your name and period output just like those used for program 5B in a function named displayMyInfo.
- 4) Implement the function named menu returning the user’s choice.
- 5) Implement the function named controlMenuExecution which consists of “if” and “elif” statements to call the other functions according to the user’s choice in the function menu.

- 6) Create functions for all of the other menu choices. Return the dictionary whenever a change is made to the dictionary in that function. Therefore, functions that only display elements in the function don't need return statements.
- 7) While much of the code can be found in our textbook, remind me to teach the class how to "invert a dictionary" for the reverse number lookup function. Since dictionaries can only lookup "keys" and not "values", we have to invert the dictionary into another dictionary called `invertedPhoneBook` in order to invert the keys and values (making what used to be keys as values and what used to be values as keys). To keep this a simple inversion, make sure that each name and phone number in the original dictionary is unique (no repeated names or phone numbers). It can be done otherwise, but the duplicate values would be grouped in lists.
- 8) In the two functions that display the names only, and the names and numbers in alphabetical order, the dictionaries need to be assigned to a list which can be sorted and displayed using a "for loop".
- 9) Remember that keys in a dictionary need to be immutable types, so be sure to put quotes around the names and phone numbers (so we can invert it) to treat them as strings.
- 10) Make sure you have DocStrings for each function including a brief description of the function, the type contract, and some sample function calls with sample expected outputs.
- 11) Save your program as `LastNameFirstNameP5D.py`.