**Work sheet for python**

1. In a company **XYZ** there are employees called **commissioned-salaried employee**which is paid their salary based on the **fixed salary plus percent they sold**. Suppose that there is a list which contains the name, sell, percent and fixed \_salary of 6 employee in this company. The program uses a dictionary, where **the key** is the **name** of employee and **value** is a list containing sell and percent and Fixed\_salary of employee, like this **{“seid”: [10000, 0.05,5000],”chala”: [2000,0.05,5000] …}.**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Sell | percent | Fixed\_salary |
| Seid | 10000 | 0.05 | 5000 |
| Chala | 2000 | 0.05 | 5000 |
| Abenezer | 4000 | 0.05 | 5000 |
| Abel | 20000 | 0.05 | 5000 |
| Kebede | 8000 | 0.05 | 5000 |
| Mohammed | 5000 | 0.05 | 5000 |

1. Calculate the salary of each employee **(salary=fixed \_ salary+(sell\*percent))** and append them to your original lists inside the dictionary: Similar to {“seid”: [10000, 0.05,5000,5500],”chala”: [2000,0.05,5000,5100] …}.
2. Print the dictionary in the following format

**Name total salary**

Seid 5500

chala 5100

abenezer 5200

abel 6000

kebede 5400

mohammed 5250

1. Print top 3 employee which is highly paid
2. Print total salary >5500
3. Write a python program that check a word from a file is **palindrome or not**
4. Write a python program that read the following word in reverse order

Name=”I love python” after reverse name=”nohtypevol I”

1. What is the output of the following program

str1="hello"

str2="world"

str1 +=str2

print str1

print len(str1)

print "helloworld" in str1

print str1.split()

1. What is the output of the following program

from math import \*

defsquare\_root(a,b=2,c=4):

x=sqrt(a\*b\*c)

print x

square\_root(2)

square\_root(2,8)

square\_root(1,2,2)

1. What is the output of the following program?

b=2

defcalc():

for a in range(1,20,4):

if a%b==0:

print a\*a

else:

print a\*b

calc()

1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Sample String : 'university'  
Expected Result : 'unty'  
Sample String : 'un'  
Expected Result : 'unun'

1. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.
2. Write a Python function that takes a list of words and returns the length of the longest one.
3. Write a Python program to get the largest number from a list.
4. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.

Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]  
Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

1. Write a Python program to remove duplicates from a list.
2. What is the output of the following program?

c = 1

def foo():

c = 0

c += 1

print c

def bar():

global c

c += 1

print c

foo()

1

foo()

1

foo()

1

bar()

2

bar()

3

1. Draw the output of the following python program on the canvas.

from cs1graphics import \*

can=Canvas(300,350)

defdraw\_face():

c=Circle(40,Point(170,170))

b=Path(Point(170,210),Point(170,280))

paper.add(c)

paper.add(b)

defdraw\_leg():

leg=Path(Point(170,280),Point(170,320))

return leg

theleg=draw\_leg()

leftleg=theleg.clone()

rightleg=theleg.clone()

leftleg.rotate(90)

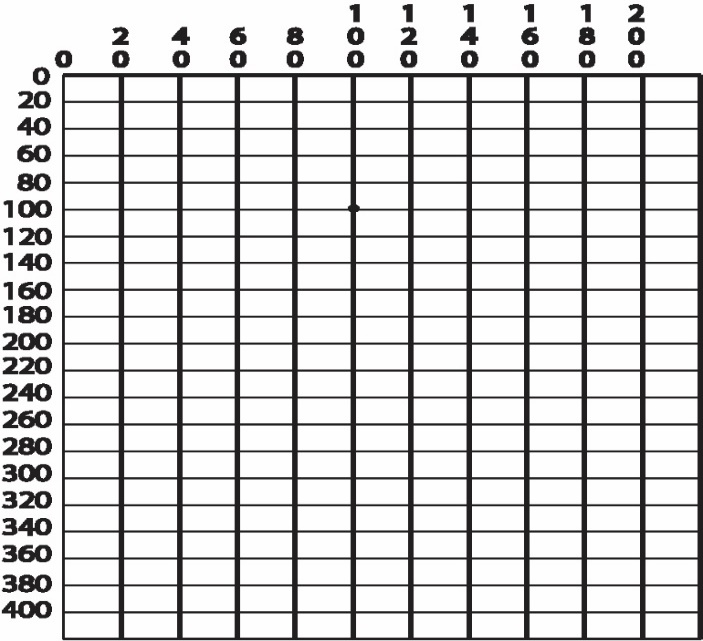
rightleg.rotate(-90)

paper.add(theleg)

paper.add(leftleg)

paper.add(rightleg)

draw\_face()

1. ****Draw the output of the following python program on the canvas.

from cs1graphics import \*

can=Canvas (400,500) path=Polygon(Point(170,170),Point(120,220),Point(220,220))

pat=Path (Point (145,320), Point(195,320))

can.add(pat)

can.add(path)

pat2=pat.clone()

pat2.rotate(-90)

can.add(pat2)

pat2.scale(2)

pat3=pat2.clone()

pat3.moveTo(195,320)

can.add(pat3)

1. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Sample Dictionary   
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

1. This program is for reflecting an object in cs1media

from cs1media import \*

statue = load\_picture("photos/statue1.jpg")

def reflection(img):

w, h = img.size()

for y in range(0,h):

for x in range(0, w/2):

pl = img.get(x, y)

pr = img.get(w-1-x, y)

img.set(x, y, pr)

img.show()

reflection(statue)

1. Write a python code that prompts a user to enter names and grades of 5 students, your program then writes the names and grade of these students into a file called “studentsgrade.txt”
2. Write a function named ***perfectsquire*** that check a number is the perfect number or not in the following list element

**perfect= [4,6,9,10,16,20]**

***NB***: perfect square is an integer that is equal to some integer squared.

***Hint***: use ***sqrt*** function that defined in math

1. Write a program that accepts two strings from the keyboard and check the length of each sting and displays true if length of the first string is equal to the second string otherwise false?
2. **For the following questions use words.txt as a working file** 
   1. Write a program to read first n lines of a file.
   2. Write a program to read last n lines of a file.
   3. Write a program to read a file line by line and store it into a list.
   4. Write a program to find the longest words.
   5. Write a program to count the number of lines in a text file
   6. Write a program to copy the contents of a file to another mywords.txt file
3. Write a program to remove an empty tuple(s) from a list of tuples.

Sample data: [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]  
Expected output: [('',), ('a', 'b'), ('a', 'b', 'c'), 'd']

1. Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary :   
dic1={1:10, 2:20}   
dic2={3:30, 4:40}   
dic3={5:50,6:60}  
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

1. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).   
   Sample Dictionary ( n = 5) :   
   Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
2. Write a program to sum all the items in a dictionary.
3. Write a program to multiply all the items in a dictionary.
4. Write a program to get the maximum and minimum value in a dictionary.
5. Write a program to sum all the items in a list.
6. Write a program to multiplies all the items in a list.
7. Write a program to get the largest number from a list.
8. Write a program to get the smallest number from a list.
9. Write a program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.  
   Sample List : ['abc', 'xyz', 'aba', '1221']  
   Expected Result : 2
10. Write a program to clone or copy a list.
11. Write a program to find the list of words that are longer than n from a given list of words.
12. Write a Python function that takes two lists and returns True if they have at least one common member.
13. Write a program to convert a list of characters into a string.
14. Write a program to append the second list on the first list.
15. Write a program to find the second smallest number in a list.
16. Write a program to get the frequency of the elements in a list.
17. Write a program to convert a list of multiple integers into a single integer

Sample list: [11, 33, 50]  
Expected Output: 113350

1. Write a program to insert an element before each element of a list.
2. Write a program to convert list to list of dictionaries.   
   Sample lists: ["Black", "Red", "Maroon", "Yellow"], ["#000000", "#FF0000", "#800000", "#FFFF00"]  
   Expected Output: [{'color\_name': 'Black', 'color\_code': '#000000'}, {'color\_name': 'Red', 'color\_code': '#FF0000'}, {'color\_name': 'Maroon', 'color\_code': '#800000'}, {'color\_name': 'Yellow', 'color\_code': '#FFFF00'}]
3. Write your own user defined function to calculate the length of a string.
4. Write a program to count the number of characters (character frequency) in a string.   
   Sample String : google.com'  
   Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}
5. Write a program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.  
   Sample String : 'abc'  
   Expected Result : 'abcing'   
   Sample String : 'string'  
   Expected Result : 'stringly'
6. Write a Python function that takes a list of words and returns the length of the longest one.
7. Write a program to remove the nth index character from a nonempty string.
8. Write a program to count the occurrences of each word in a given sentence.
9. Write a program to create a Caesar encryption.

Note : In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

1. Write a program to reverse a string.
2. Write a user-defined function to convert a string in a list.
3. Write a program to count and display the vowels of a given text.
4. Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password.
5. Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right.

* Keep the game going until the user types “exit”
* Keep track of how many guesses the user has taken, and when the game ends, print this out.

1. Suppose you have the following list of numbers to sort:

[19, 1, 9, 7, 3, 10, 13, 15, 8, 12] what will be the partially sorted list after three complete passes (rounds) in case

* selection sort,
* bubble sort