A picture containing text, metalware, gear

Description automatically generated



**LAB: Artificial Intelligence**

|  |  |
| --- | --- |
| **Lab Instructor** | Ms. Saba Aslam |
| **Department** | Computer Science |

**Python Lab : Python Basics to Advance**

**Instructions**

1. Make a word document paste your code and output there.

2. Comments in the code explaining chunks of the code is important.

3. Plagiarism is strictly prohibited, 0 marks would be given to students who cheat.  4. Late submissions are not allowed.

**Task 1**

Write a python code to print the following sequence of string.

“My name is ABC,

My roll number is 18F-XXXX,

I Like AI Subject :D ”

***Note: Name and roll number should be yours.***

**CODE:**

print("My name is Wajid Ali \nMy roll number is 19F-0313 \nI Like AI Subject :D ")

**OUTPUT:**

Text

Description automatically generated

**Task 2**

Write a python code to do the following task

i. Take 2 integers from user input

ii. Make the following functions of Add, Subtract, Multiply and Divide Remainder, Square,  and Cube of a given number.

iii. Display the output of each function.

***Note: Square and Cube functions can take 1 value.***

**CODE:**

def ADD(x,y):  
 return x+y  
  
def SUB(x,y):  
 return x-y  
  
def MUL(x,y):  
 return x\*y  
  
def DIV(x,y):  
 return x/y  
def SQR(x):  
 return 4\*x  
def CUBE(x):  
 return x\*x\*x  
  
while True:  
 print('\nEnter 1 for addition')  
 print('Enter 2 for subtraction')  
 print('Enter 3 for multiplication')  
 print('Enter 4 for division')  
 print('Enter 5 for square')  
 print('Enter 6 for cube')  
  
 select = int(input(' Enter option : '))  
 if(select<=6):  
 number1 = int(input('\nEnter value of first number : '))  
 if (select <= 4):  
 number2 = int(input('\nEnter value of second number : '))  
  
 if select == 1:  
 print('Sum : ')  
 print(ADD(number1, number2))  
  
 elif select == 2:  
 print('Substraction : ')  
 print(SUB(number1, number2))  
  
 elif select == 3:  
 print('Multiply : ')  
 print(MUL(number1, number2))  
  
 elif select == 4:  
 print('Division : ')  
 print(DIV(number1, number2))  
  
 elif select == 5:  
 print('Square : ')  
 print(SQR(number1))  
  
 elif select == 6:  
 print('Cube : ')  
 print(CUBE(number1))  
 else:  
 print('Invalid option')

**OUTPUT:**

Text

Description automatically generated

**Task 3**

Write a python code to do the following task

i. Take 3 float values from user input

ii. Make a function that adds 5 to first value 10 to second value and 15 to third value iii. Return these 3 values and show it in the main function.

**CODE:**

def ADD(x,y,z):  
 a = float(x+5)  
 b = float(y+10)  
 c = float(z+15)  
 print('\nValue of first number after adding 5 : ', end=" ")  
 print(a)  
 print('\nValue of second number after adding 10 : ', end=" ")  
 print(b)  
 print('\nValue of third number after adding 15 : ', end=" ")  
 print(c)  
  
def main():  
 number1 = float(input('\nEnter value of first number : '))  
 number2 = float(input('\nEnter value of second number : '))  
 number3 = float(input('\nEnter value of third number : '))  
 print(ADD(number1,number2,number3))  
  
main()

**OUTPUT:**

Text

Description automatically generated

**Task 4**

Write a python code to do the following task

i. Take 2 strings with sequence = “HELLO, WORLD” and “HOW, arE yOu”.  ii. Do the following tasks Concatenate the 2 strings, Make the 2 strings in upper and  lowercase, Slice the strings using python string slicing. The following string should  look like this after slicing; “,World” and “EyO”.

iii. For the second string you have to use only negative number for slicing.

***Note: Read the instructions carefully. And the output of string should be exact especially in  slicing.***

**CODE:**

str1 = 'HELLO, WORLD'  
str2 = 'HOW, arE yOu'  
  
print('\nConcatenate the strings :', end=" ")  
print(str1 + str2)  
  
print('\nFirst string in lower case :',end=" ")  
print(str1.lower())  
  
print('Second string in lower case :',end=" ")  
print(str2.lower())  
  
print('\nFirst string in upper case :',end=" ")  
print(str1.upper())  
  
print('Second string in upper case :',end=" ")  
print(str2.upper())  
  
print('\nFirst string slicing :',end=" ")  
print(str1[5:])  
print('Second string slicing :',end=" ")  
print(str2[7:11])  
  
print('\nFirst string slicing using negative numbers :',end=" ")  
print(str1[-7:])  
print('Second string slicing using negative numbers :',end=" ")  
print(str2[-5:-1])

**OUTPUT:**

Text

Description automatically generated

**Task 5**

Take a start value and end value from user and print Even and Odd numbers between them using  **for** and **while** loop.

**CODE:**

while True:  
 print('\n\nEnter 1 for For loop')  
 print('Enter 2 for While loop')  
  
 select = int(input(' Enter option : '))  
  
 if (select == 1):  
  
 num1 = int(input('\nEnter value of first number : '))  
 num2 = int(input('\nEnter value of second number : '))  
  
 print('\nEVEN numbers are :', end=" ")  
 for number in range(num1, num2):  
 if (number % 2 == 0):  
 print(number, end=" ")  
  
 print('\nODD numbers are :', end=" ")  
 for number in range(num1, num2):  
 if (number % 2 != 0):  
 print(number, end=" ")  
  
 elif (select == 2):  
 num1 = int(input('\nEnter value of first number : '))  
 num2 = int(input('\nEnter value of second number : '))  
 num3 = num1  
  
 print('\nEVEN numbers are :', end=" ")  
 while num1 < num2:  
 if (num1 % 2 == 0):  
 print(num1, end=" ")  
 num1 = num1 + 1  
  
 print('\nODD numbers are :', end=" ")  
 while num3 < num2:  
 if (num3 % 2 != 0):  
 print(num3, end=" ")  
 num3 = num3 + 1

**OUTPUT:**

Text

Description automatically generated

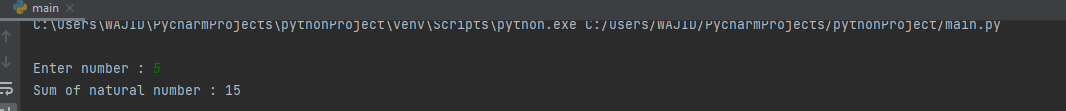
**Task 6**

Get number from the user and find the sum of natural numbers till that number.

**CODE:**

num1 = int(input('\nEnter number : '))  
sum = int(num1/2\*(num1+1))  
print('Sum of natural number :',end=" ")  
print(sum)

**OUTPUT:**



**Task 7**

Get number from the user and write a program to reverse the number.

**CODE:**

num1 = int(input("\nEnter Number : "))  
Reverse = 0  
while (num1 > 0):  
 Reminder = num1 % 10  
 Reverse = (Reverse \* 10) + Reminder  
 num1 = num1 // 10  
  
print("\nReverse of given number is :",end=" ")  
print(Reverse)

**OUTPUT:**

Text

Description automatically generated

**Task 8**

Use slicing and string concatenation. Str = '''Master has failed more, than the beginner has tried''

**CODE:**

str1 = 'Master has failed more,'  
str2 = ' than the beginner has tried'  
print('\nString concatenation :',end=" ")  
print(str1 + str2)  
  
print('\nstring slicing :', end=" ")

print(str1[2:14])

**OUTPUT:**

Text

Description automatically generated

**Task 9**

list = ["apple", "cherry", "orange", "kiwi", "melon", "mango"]

i. Remove “cherry” and “melon” from list.

ii. Add “banana” at second last index.

**CODE:**

list = ["apple", "cherry", "orange", "kiwi", "melon", "mango"]  
list.remove("cherry")  
list.remove("melon")  
print('\nList after removing elements : ',end=" ")  
print(list)  
  
list.insert(-1,'banana')  
print('\nList after adding elements : ',end=" ")  
print(list)

**OUTPUT:**

Graphical user interface, text

Description automatically generated

**Task 10**

list = [ 1, 4, 56, 2, 4 , 12, 6, 89 ,11, 0]

Write a program to empty list using **while loop** and **pop()** function.

**CODE:**

list = [ 1, 4, 56, 2, 4 , 12, 6, 89 ,11, 0]  
i = 0  
while i < len(list):  
 print('After pop :', end=" ")  
 print(list.pop())

**OUTPUT:**

Text

Description automatically generated

**Task 11**

marks = {'AI':74,'CN':76,'DS':42,'PS':54}

i. Print the sum of marks using **for loop**.

ii. Print keys and values both using **for loop**.

iii. Print max marks and relative subject name.

iv. Print mix marks and relative subject name.

**CODE:**

marks = {'AI':74,'CN':76,'DS':42,'PS':54}  
sum=0  
for i in marks:  
 sum = sum + marks[i]  
print('\nSum of marks :',end=" ")  
print(sum)  
  
print('\nKEYS AND VALUES OF DICTIONARY')  
for i,j in marks.items():  
 print('\tKEY :',end=" ")  
 print(i,end=" ")  
 print('Values :', end=" ")  
 print(j)  
  
max=0  
for i,j in marks.items():  
 if max <= marks[i]:  
 max = marks[i]  
 thisdict = i  
  
print('\nMAXIMUM MARKS is : ',max,' \nSUBJECT NAME iS : ',thisdict)  
  
  
min=max  
for i,j in marks.items():  
 if min >= marks[i]:  
 min = marks[i]  
 thisdict = i  
  
print('\nMINIMUM MARKS is : ',min,' \nSUBJECT NAME iS : ',thisdict)

**OUTPUT:**

Text

Description automatically generated

**Task 12**

Write a python code to find if the number is an Armstrong number or not!  ***Note: abcd = an + bn + cn +dn +…., n is the total number of digits.***

**CODE:**

num = int(input("\nEnter a number : "))  
power = len(str(num))  
sum = 0  
temp = num  
  
while temp > 0:  
 digit = temp % 10  
 sum = sum + digit \*\* power  
 temp = temp // 10  
  
if num == sum:  
 print(num, "is an Armstrong number")  
else:  
 print(num, "is not an Armstrong number")

**OUTPUT:**

Text

Description automatically generated

**Task 13**

Write a python code to find the factorial of number using recursion.

**CODE:**

num = int(input('\nEnter Number : '))  
def factorial(n):  
 if (n == 1 or n == 0):  
 return 1  
 else:  
 return (n \* factorial(n - 1))  
  
print("Factorial of given number is : ",factorial(num))

**OUTPUT:**

Text

Description automatically generated

**Task 14**

Write a python code to make a list. Populate it with random number using random number  generator and then find the min and max value from the list. Size of list should be given by user.  ***Note: do not use max and min built-in function.***

**CODE:**

import random  
rand\_list = []  
num = int(input('\nEnter range of randon number : '))  
start= int(input('\nEnter starting value of randon number : '))  
end = int(input('Enter ending value of randon number : '))  
for i in range(num):  
 rand\_list.append(random.randint(start, end))  
print(rand\_list)  
  
max=0  
for i in range(num):  
 if max <= rand\_list[i]:  
 max = rand\_list[i]  
print('\nMaximum of the list is : ',max)  
  
min=max  
for i in range(num):  
 if min >= rand\_list[i]:  
 min = rand\_list[i]  
print('Minimum of the list is : ',min)

**OUTPUT:**

Text

Description automatically generated

**Task 15**

Write a python code to create a tuple. And do the following tasks.

i. Create Tuple =

(“john”,”mark”,12,”14”,”orange”,4.5)

ii. Then update the Tuple and add a new value 6.5.

iii. After updating the value, create a function that

shows the count of how many integers, strings and

float variables does it have.

**CODE:**

Tuple = ('john','mark',12,14,'orange',4.5)  
print('\nTuple : ',Tuple)  
  
x = list(Tuple)  
x.append(6.5)  
Tuple = tuple(x)  
print('\nTuple after add item : ',Tuple)  
  
  
def counter():  
 str\_count =0  
 int\_count = 0  
 float\_count=0  
 for i in range(len(Tuple)):  
 if type('sd')==type(Tuple[i]):  
 str\_count=str\_count + 1  
  
 elif type(1)==type(Tuple[i]):  
 int\_count=int\_count + 1  
  
 elif type(1.99) == type(Tuple[i]):  
 float\_count = float\_count + 1  
  
 print('\nTotal string variables in tuple : ',str\_count)  
 print('Total int variables in tuple : ',int\_count)  
 print('Total float variables in tuple : ',float\_count)  
counter()

**OUTPUT:**

Text

Description automatically generated

**Task 16**

Write a python code to create a dictionary. And do the following tasks.

i. Create Dictionary = {“brand”:”Samsung”,”OS

type”:”Oreo”,”color”:”black”,”camera”:”42 megapixels”,”year”:2012}

ii. Then add a list in the current dictionary with key = “sizes” and values of random  numbers.

iii. Then delete the “year” key from the dictionary.

iv. Lastly show the dictionary in following order. Use loop to show the dictionary:  a. Brand

b. Color

c. Camera

d. OS-type

e. Sizes

**CODE:**

Dictionary = {'Brand':'Samsung','OS-type':'Oreo','Color':'black','Camera':'42 megapixels','Year': 2012}  
  
print('\nGiven dictionary is : ',Dictionary)  
  
Dictionary['sizes']=[12,23,45]  
print('\nDictionary after adding sizes is : ',Dictionary)  
  
del Dictionary['Year']  
print('\nDictionary after deleting year is : ',Dictionary)  
  
print('\nDictionary in order : ')  
for i in sorted(Dictionary.keys()):  
 print(i)

**OUTPUT:**

Text

Description automatically generated

**Task 17**

Create a function that takes list list1 and list2 as parameter, create a list list3 inside the function  and store the sum of list list1, list2 into list3 then return list3.

list1 = [11,22,33,44,21,54,67,54,33,222,4]

list2 = [3,4,5,32,21,33,66,75,87,97,1]

**CODE:**

list1 = [11,22,33,44,21,54,67,54,33,222,4]  
list2 = [3,4,5,32,21,33,66,75,87,97,1]  
print('\nList1 is : ',list1)  
print('List2 is : ',list2)  
  
sum=0  
def sum\_lists(list1,list2):  
 list3=[]  
 for i in range(len(list1)):  
 sum=int(list1[i]+list2[i])  
 list3.append(sum)  
 print('\nSum of list1 and list2 is : ',list3)  
sum\_lists(list1,list2)

**OUTPUT:**

Text

Description automatically generated

**Task 18**

Create a function dsort() to sort a list in descending order, taking a list as argument and returning  it. Use the following list and print it.

list = [ 5, 6, 7, 23 ,12 ,3, 3, 4 ,5, 12, 34]

**CODE:**

list1 = [ 5, 6, 7, 23 ,12 ,3, 3, 4 ,5, 12, 34]  
print('\nGiven list : ',list1)  
def dsort(list):  
 list1.sort(reverse=True)  
 print('\nList in Descending Order : ', list1)  
dsort(list1)

**OUTPUT:**

Text

Description automatically generated

**Task 19**

Implement Stack and Queue in Python

Write separate functions for inserting and deleting and ask the user to input the values.

**CODE:**

1. **For stack:**

def create\_stack():  
 stack = []  
 return stack  
  
def check\_empty(stack):  
 return len(stack) == 0  
  
def push(stack, item):  
 stack.append(item)  
 print("pushed item : " , item)  
  
def pop(stack):  
 if (check\_empty(stack)):  
 return "stack is empty"  
  
 return stack.pop()  
  
stack = create\_stack()  
push(stack, str(1))  
push(stack, str(2))  
push(stack, str(3))  
push(stack, str(4))  
print("\npopped item : ",pop(stack))  
print("\nstack after popping an element: " ,str(stack))

**OUTPUT:**

Text

Description automatically generated

1. **For queue:**

**CODE:**

q=[]  
size=int(input("\nEnter the size of Queue : "))  
  
def Enqueue():  
 if len(q)==size:  
 print("\nQueue is Full")  
 else:  
 element=input("\nEnter the element : ")  
 q.append(element)  
 print(element,"is added to the Queue")  
def dequeue():  
 if not q:# or if len(stack)==0  
 print("\nQueue is Empty add Element for pop")  
 else:  
 e=q.pop(0)  
 print("\nElement removed : ",e)  
def display():  
  
 while True:  
 print("\nSelect the Operation:\n1 for Add \n2 for Delete\n3 for Display\n4 for Quit")  
 choice=int(input('Select option : '))  
 if choice==1:  
 Enqueue()  
 elif choice==2:  
 dequeue()  
 elif choice==3:  
 print(q)  
 elif choice==4:  
 break  
 else:  
 print("Invalid Option!!!")  
display()

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

**Task 20**

Write a python program to create a class. And do the following tasks

i Create a class named Student with following attributes. First-name, Last-name, Age, CNIC,  Courses (list), Gender, CGPA, SGPA, Current\_Credit\_Hours.

ii Create a constructor for making new instances of the class.

iii Getter, Setter method should be included as well

iv Also add CRUD operation methods in the class.

Create at least 3 objects to show the working of the class.

**Task 21**

From Task 20 you have to create lists of objects, dictionary of objects and tuples of objects.  Compare these objects if they are equal or not. Also print each objects values from different data-structures.

**Task 22**

Write a python program to create a class. And do the following tasks

i Make a class named Shape, Rectangle and Circle.

ii In the class Shape there is a print function that tells which object is calling it and prints the class  name.

iii Also make 2 functions of area and perimeter in Rectangle and Circle class.

iv Display the name of both classes using the print function in class Shape and also the calculations  of each function of their respective classes.

***Note: You cannot use Shape object.***

***Hint: Inheritance!***

**Task 23**

Plot the given graph

Note: Use matplotlib or any other python library for plotting graph.

Diagram

Description automatically generated

**CODE:**

import networkx as nx  
import matplotlib.pyplot as plt  
  
G = nx.Graph()  
G.add\_edges\_from([('Spider Man', 'Civil War'), ('Spider Man', 'Iron Man'), ('Spider Man', 'AVENGERS'),  
 ('Iron Man', 'Civil War'),('Iron Man', 'AVENGERS'),  
 ('Civil War', 'Captain America'), ('Civil War', 'Ant Man'),  
 ('Ant Man', 'Captain America'),('Ant Man', 'AVENGERS'),  
 ('Captain America', 'AVENGERS'),  
 ('AVENGERS', 'Thor'), ('AVENGERS', 'Hulk'),  
 ('Thor', 'Ragnarok'), ('Hulk','Ragnarok')])  
  
plt.figure(figsize=(25,25))  
nx.draw\_networkx(G, with\_labels=True, node\_color='lightblue',node\_size =5000)  
plt.draw()  
plt.show()

**OUTPUT:**

Diagram

Description automatically generated

**Task 24**

Implement given graph using adjacency list

Shape, arrow

Description automatically generated

**CODE:**

class AdjNode:  
 def \_\_init\_\_(self, value):  
 self.vertex = value  
 self.next = None  
  
class Graph:  
 def \_\_init\_\_(self, num):  
 self.V = num  
 self.graph = [None] \* self.V  
  
 def add\_edge(self, s, d):  
 node = AdjNode(d)  
 node.next = self.graph[s]  
 self.graph[s] = node  
  
 node = AdjNode(s)  
 node.next = self.graph[d]  
 self.graph[d] = node  
  
 # Print the graph  
 def print\_agraph(self):  
 for i in range(self.V):  
 print("Vertex " + str(i) + ":", end="")  
 temp = self.graph[i]  
 while temp:  
 print(" -> {}".format(temp.vertex), end="")  
 temp = temp.next  
 print(" \n")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 V = 9  
  
 #a=1 b=2 c=3 d=4 e=5 f=6 g=7 h=8  
  
 graph = Graph(V)  
 graph.add\_edge(1, 2)  
 graph.add\_edge(1, 3)  
 graph.add\_edge(1, 4)  
 graph.add\_edge(2, 5)  
 graph.add\_edge(2, 6)  
 graph.add\_edge(5, 7)  
 graph.add\_edge(6, 8)  
  
 graph.print\_agraph()

**OUTPUT:**

Text

Description automatically generated

**Task 25**

Implement the given graph using adjacency matrix

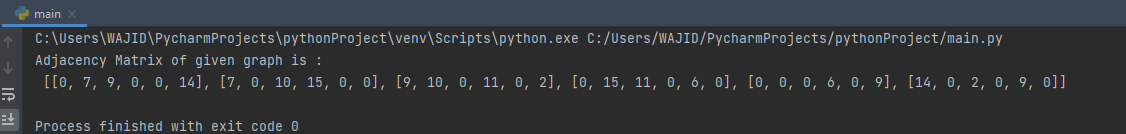
A picture containing watch

Description automatically generated

**CODE:**

def add\_vertex(v):  
 global graph  
 global vertices\_no  
 global vertices  
 if v in vertices:  
 print("Vertex ", v, " already exists")  
 else:  
 vertices\_no = vertices\_no + 1  
 vertices.append(v)  
 if vertices\_no > 1:  
 for vertex in graph:  
 vertex.append(0)  
 temp = []  
 for i in range(vertices\_no):  
 temp.append(0)  
 graph.append(temp)  
  
  
def add\_edge(v1, v2, e):  
 global graph  
 global vertices\_no  
 global vertices  
  
 if v1 not in vertices:  
 print("Vertex ", v1, " does not exist.")  
  
 elif v2 not in vertices:  
 print("Vertex ", v2, " does not exist.")  
  
 else:  
 index1 = vertices.index(v1)  
 index2 = vertices.index(v2)  
 graph[index1][index2] = e  
  
vertices = []  
  
vertices\_no = 0  
graph = []  
  
add\_vertex(1)  
add\_vertex(2)  
add\_vertex(3)  
add\_vertex(4)  
add\_vertex(5)  
add\_vertex(6)  
  
add\_edge(1, 2, 7)  
add\_edge(1, 6, 14)  
add\_edge(1, 3, 9)  
add\_edge(2, 4, 15)  
add\_edge(2, 3, 10)  
add\_edge(2, 1, 7)  
add\_edge(3, 1, 9)  
add\_edge(3, 2, 10)  
add\_edge(3, 4, 11)  
add\_edge(3, 6, 2)  
add\_edge(4, 3, 11)  
add\_edge(4, 2, 15)  
add\_edge(4, 5, 6)  
add\_edge(5, 4, 6)  
add\_edge(5, 6, 9)  
add\_edge(6, 3, 2)  
add\_edge(6, 1, 14)  
add\_edge(6, 5, 9)  
  
print("Adjacency Matrix of given graph is : \n", graph)

**OUTPUT:**



**Pycharm**

**Note: For this part, provide only screenshots.**

**Task 26**

**Write some basic shortcuts of pycharm (minimum 5) relating to code indentation, changing theme,  etc. (no screenshots needed)**

1. **alt+ctrl+I** Indent current line or selected block according to the code style settings

.

1. **ctrl+/** Comment/uncomment current line or selected block with line comments
2. **alt+ctrl+L** Reformat code
3. **shift+F10** Run
4. **shift+F9** Debug
5. **alt+5** Activate Debug window
6. In the **Settings/Preferences** dialog (**Ctrl+Alt+S)**, select **Appearance & Behavior | Appearance**.

Select the UI theme from the **Theme** list.

**Task 27**

**How to install library from terminal**

Text

Description automatically generated

**How to install library without using terminal**

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

**Task 28**

**How to zoom in pycharm**

**ANSWER**

Enable scroll-zoom in **PyCharm**:

1. Go to **File > Settings** to open the **Settings** dialog
2. In the **Settings** dialog, navigate to **Editor > General**
3. Check the **Change font size (Zoom) with Ctrl + Mouse Wheel** option

Text

Description automatically generated

**Task 30**

**How to change theme in Pycharm**

**ANSWER**

In the **Settings/Preferences** dialog (**Ctrl+Alt+S)**, select **Appearance**

**Behavior | Appearance**.

Select the UI theme from the **Theme** list.

