In [1]:

```
!dir
```

```
 Volume in drive D is Arjun
 Volume Serial Number is 3A19-3650

 Directory of D:\SRET\FirstYear\PythonProgramming\Class\Revision4\Code

02-11-2022  14:41    <DIR>          .
02-11-2022  14:09    <DIR>          ..
02-11-2022  14:41    <DIR>          .ipynb_checkpoints
02-11-2022  14:41                72 Untitled.ipynb
               1 File(s)             72 bytes
               3 Dir(s)  154,625,036,288 bytes free
```

phone number credit card mail id date sub method module 5

Module1 - Module5

Module1

In [2]:

```
for i in range(1,6):
    print()
    for j in range(1,i+1):
        print(chr(ord('A')+j-1),end =" ")

```

```

A 
A B 
A B C 
A B C D 
A B C D E 
```

In [3]:

```
for i in range(1,6):
    print()
    for j in range(1,i+1):
        print(j , end = ' ')
```

```

1 
1 2 
1 2 3 
1 2 3 4 
1 2 3 4 5 
```

In [4]:

```
for i in range(1,6):
    print()
    for j in range(1,i+1):
        print("1",end = ' ')
```

```
1
1 1
1 1 1
1 1 1 1
1 1 1 1 1
```

In [5]:

```
for i in range(1,6):
    print()
    for j in range(1,i+1):
        print('#',end = " ")
```

```
#
# #
# # #
# # # #
# # # # #
```

In [6]:

```
#rev number
num = int(input("Enter a number: "))
temp = num
rev = 0
while num>0:
    dig = num%10
    rev = rev*10+dig
    num = num//10
if temp == rev:
    print("Palindrome")
else:
    print("Nit a palindrome")
```

```
Enter a number: 1234321
Palindrome
```

In [7]:

```python
#Armstrong number
num = int(input("Enter a number: "))
temp = num
sum_val = 0
while num>0:
    dig = num%10
    sum_val = sum_val+dig**3
    num = num//10
if sum_val == temp:
    print("Armstrong Number")
else:
    print("Not a Armstrong Number")
```

```
Enter a number: 153
Armstrong Number
```

In [8]:

```python
#Two digit special num
#rev number
num = int(input("Enter a number: "))
temp = num
sum_val = 0
prod_val = 1
while num>0:
    dig = num%10
    sum_val += dig
    prod_val *= dig
    num = num//10
res = sum_val+prod_val
if res == temp:
    print("Two digit special number")
else:
    print("Not  a two digit special number")
```

```
Enter a number: 59
Two digit special number
```

In [9]:

```
#if else
name = input("Enter name: ")
exp = int(input('Enter no of years of experienece: '))
sal = int(input('Enter salary: '))
if sal<10000:
    print("Name: ",name)
    print("Bonus: ",0.12*sal)
    print("Net salary: ",sal+0.12*sal)
elif sal>10000 and exp<2:
    print("Name: ",name)
    print("Bonus: ",0.02*sal)
    print("Net salary: ",sal+0.12*sal)
elif sal>10000 and exp>5:
    print("Name: ",name)
    print("Bonus: ",0.13*sal)
    print("Net salary: ",sal+0.13*sal)
else:
    print("No Bonus")


```

```
Enter name: Arjun
Enter no of years of experienece: 6
Enter salary: 10001
No Bonus
```

In [10]:

```
#pass
for i in range(9):
    print(i)
    pass
```

```
0
1
2
3
4
5
6
7
8
```

In [11]:

```
#continue
for i in range(9):
    if i==5:
        continue
    print(i)
```

```
0
1
2
3
4
6
7
8
```

In [12]:

```
#break
for i in range(9):
    if i==6:
        break
    print(i)
```

```
0
1
2
3
4
5
```

Module2

In [166]:

```
lst = [1,5,9,78,56]
```

In [167]:

```
lst.sort()
```

In [168]:

```
lst
```

Out[168]:

```
[1, 5, 9, 56, 78]
```

In [169]:

```
lst.append(78)
lst
```

Out[169]:

```
[1, 5, 9, 56, 78, 78]
```

In [170]:

```
lst.count(78)

```

Out[170]:

```
2
```

In [18]:

```
lst.insert(3,78)
lst
```

Out[18]:

```
[1, 5, 9, 78, 56, 78, 78]
```

In [19]:

```
lst.pop(1)
lst
```

Out[19]:

```
[1, 9, 78, 56, 78, 78]
```

In [20]:

```
lst.reverse()
lst
```

Out[20]:

```
[78, 78, 56, 78, 9, 1]
```

In [21]:

```
lst
```

Out[21]:

```
[78, 78, 56, 78, 9, 1]
```

In [22]:

```
lst[::-1]
```

Out[22]:

```
[1, 9, 78, 56, 78, 78]
```

In [23]:

```
lst.remove(1)
lst
```

Out[23]:

```
[78, 78, 56, 78, 9]
```

In [24]:

```
lst.sort()
```

In [25]:

```
lst
```

Out[25]:

```
[9, 56, 78, 78, 78]
```

In [26]:

```
lst = set(lst)
lst
```

Out[26]:

```
{9, 56, 78}
```

In [27]:

```
tuple = tuple(lst)
```

In [30]:

```
tuple
```

Out[30]:

```
(56, 9, 78)
```

In [31]:

```
tuple.count(8)
```

Out[31]:

```
0
```

In [33]:

```
tuple.index(78)
```

Out[33]:

```
2
```

In [34]:

```
tuple[2]
```

Out[34]:

```
78
```

In [35]:

```
tuple[2:0:-1]
```

Out[35]:

```
(78, 9)
```

In [36]:

```
lst = [7,8,9]
fnl_lst = []
for i in lst:
    if i>=8:
        fnl_lst.append(i)
print("Final list: ",fnl_lst)
```

```
Final list:  [8, 9]
```

In [39]:

```
lst = []
num = int(input("Enter the no of elements: "))
for i in range(num):
    val = int(input("Enter a number to append: "))
    lst.append(val)
print("Got input from user and appended successfully")
print("List: ",lst)
```

```
Enter the no of elements: 4
Enter a number to append: 1
Enter a number to append: 7
Enter a number to append: 8
Enter a number to append: 6
Got input from user and appended successfully
List:  [1, 7, 8, 6]
```

In [40]:

```python
lst = []
num = int(input("Enter the no of elements: "))
for i in range(num):
    val = int(input("Enter a number to append: "))
    if val>100:
        lst.append("Excess")
    else:
        lst.append(val)
print("Got input from user and appended successfully")
print("List: ",lst)
```

```
Enter the no of elements: 4
Enter a number to append: 1
Enter a number to append: 100
Enter a number to append: 78
Enter a number to append: 500
Got input from user and appended successfully
List:  [1, 100, 78, 'Excess']
```

In [43]:

```python
pos_lst = []
neg_lst = []
num = int(input("Enter the no of elements: "))
for i in range(num):
    val = int(input("Enter a number to append: "))
    if val>0:
        pos_lst.append(val)
    else:
        neg_lst.append(val)
print("Positive list : ",pos_lst)
print("Negative List : ",neg_lst)
```

```
Enter the no of elements: 4
Enter a number to append: -2
Enter a number to append: -6
Enter a number to append: 1
Enter a number to append: 4
Positive list :  [1, 4]
Negative List :  [-2, -6]
```

In [45]:

```python
lst = ["Python","","","","c","Java",""]
while "" in lst:
    lst.remove("")
lst
```

Out[45]:

```
['Python', 'c', 'Java']
```

In [171]:

```python
lst = [45,56,78,89,56,89,89,23,12,89]
while 89 in lst:
    lst.remove(89)
```

In [172]:

```
lst
```

Out[172]:

```
[45, 56, 78, 56, 23, 12]
```

In [53]:

```
#string manipulatiuon
str = "Arjun"
str.upper()
```

Out[53]:

```
'ARJUN'
```

In [54]:

```
str.isupper()
```

Out[54]:

```
False
```

In [55]:

```
var = "Arjun"
var = var.upper()
var.isupper()
```

Out[55]:

```
True
```

In [56]:

```
var = var.lower()
var.islower()
```

Out[56]:

```
True
```

In [57]:

```
var.capitalize()
```

Out[57]:

```
'Arjun'
```

In [59]:

```
var.startswith("a")
```

Out[59]:

```
True
```

In [61]:

```
var = "Arjjjun"
var.endswith('n')
```

Out[61]:

```
True
```

In [62]:

```
var.count('j')
```

Out[62]:

```
3
```

In [63]:

```
var.find("n")
```

Out[63]:

```
6
```

In [64]:

```
var.index("A")
```

Out[64]:

```
0
```

In [66]:

```
var.replace("A","r")
```

Out[66]:

```
'rrjjjun'
```

In [67]:

```
#dictionary - refer ppt()
```

In [174]:

```
dic = {1:"Arjun",2:"Sathish",3:"Thenu"}
```

In [175]:

```
dic[4] = "Sundar"
```

In [176]:

```
dic
```

Out[176]:

```
{1: 'Arjun', 2: 'Sathish', 3: 'Thenu', 4: 'Sundar'}
```

In [179]:

```
print(list(dic.keys()))
print(list(dic.values()))
```

```
[1, 2, 3, 4]
['Arjun', 'Sathish', 'Thenu', 'Sundar']
```

In [182]:

```
print(dic[1])
print(dic[2])
```

```
Arjun
Sathish
```

In [183]:

```
print(dic.get(1))
```

```
Arjun
```

In [184]:

```
dic[3] = "Arjun"
```

In [185]:

```
dic
```

Out[185]:

```
{1: 'Arjun', 2: 'Sathish', 3: 'Arjun', 4: 'Sundar'}
```

In [190]:

```
1 in dic
```

Out[190]:

```
True
```

In [191]:

```
"Arjun" in dic
```

Out[191]:

```
False
```

In [193]:

```python
for k in dic:
    print("KEY: ",k)
    print("VALUE: ",dic[k])
```

```
KEY:  1
VALUE:  Arjun
KEY:  2
VALUE:  Sathish
KEY:  3
VALUE:  Arjun
KEY:  4
VALUE:  Sundar
```

In [194]:

```python
for k,v in dic.items():
    print("KEY: ",k)
    print("VALUE: ",v)
```

```
KEY:  1
VALUE:  Arjun
KEY:  2
VALUE:  Sathish
KEY:  3
VALUE:  Arjun
KEY:  4
VALUE:  Sundar
```

In [196]:

```python
word = ["Arjun","Sathish","Thenu","Sundar"]
dic1 = {}
for i in word:
    dic1[i] = len(i)
dic1
```

Out[196]:

```
{'Arjun': 5, 'Sathish': 7, 'Thenu': 5, 'Sundar': 6}
```

Module3

In [198]:

```
def cost_ride(nights):
    print("Cost Ride: ",n*1500)
    def plane_ride_cost(city):
        hrs = int(input("Enter no.of hours: "))
        sel = int(input("Choose city:\n1 for CMBT\n2 for OMR\n3 for Anna Nagar\n"))
        if sel==1:
            print("Ride cost: ",hrs*150)
        elif sel==2:
            print("Ride cost: ",hrs*250)
        elif sel==3:
            print("Ride cost: ",hrs*350)
        else:
            pass
    plane_ride_cost('hrs')
n = int(input("Enter number of nights: "))
cost_ride(n)
```

```
Enter number of nights: 5
Cost Ride:  7500
Enter no.of hours: 4
Choose city:
1 for CMBT
2 for OMR
3 for Anna Nagar
1
Ride cost:  600
```

In [75]:

```
#required arguments
def add(a,b):
    print("Adding :",a+b)
add(7,5)
```

```
Adding : 12
```

In [89]:

```
#keyword Arguments
def greet(name,dept):
    print("Name: ",name)
    print("Dept: ",dept)
greet(dept = "CYS",name = "Arjun")
```

```
Name:  Arjun
Dept:  CYS
```

In [87]:

```python
def welcome(name,dept="CYS"):
    print("My name is: ",name)
    print("My dept is: ",dept)
n = input("Enter name: ")
d = input("Enter dept: ")
if len(d) == 0:
    welcome(n)
else:
    welcome(n,d)
```

```
Enter name: Arjun
Enter dept: 
My name is:  Arjun
My dept is:  CYS
```

In [94]:

```python
#finding a word in a sentence and counting it
with open("Text.txt","w") as f:
    f.write("Hey user! this is a code in Python and tomorrow is my python final exam")
    f.close()
with open("Text.txt") as f:
    a = f.read()
    spl = a.split(" ")
    srch_msg = input('Enter the word to find the count: ')
    srch_msg = srch_msg.lower()
    word_read = 0
    for char in spl:
        char = char.lower()
        if char == srch_msg:
            word_read+=1
print("The word ",srch_msg,"is found",word_read,"times in the sentence")

```

```
Enter the word to find the count: pYthoN
The word  python is found 2 times in the sentence
```

In [202]:

```python
#finding count of vowels and percentage of vowels
with open("Text.txt","w") as f:
    f.write("Hello All")
    f.close()
with open("Text.txt") as f:
    a = f.read()
    vow_read = 0
    cons_read = 0
    space_count = 0
    for char in a:
        if char in "aeiouAEIOU":
            vow_read+=1
#         elif char == " ":
#             space_count+=1
        else:
            cons_read+=1
    vow_perc = 0
    cons_perc = 0
    space_perc = 0
    vow_perc = vow_read*100//len(a)
    cons_perc = cons_read*100//len(a)
    space_perc = space_count*100//len(a)
    print("Vowel Percentage: ",vow_perc)
    print("Consonant Percentage: ",cons_perc)
#     print("Space percentage: ",space_perc)
```

```
Vowel Percentage:  33
Consonant Percentage:  66
```

```
#Exception Handling


EXCEPTION HANDLING:
exceptions are raised when the program is syntactically correct but the code is
interrupted and results in error


ERROR
Errors are the problems in a program due to which the program will stop the
execution.two types, run time & logical.


Types of error:
1.Run time error - syntax correct , logic is wrong poor understanding of the logic.logical error can be handled by program
2.Compile time error, syntax error - poor understanding of the program. We can’t handle the syntax error.


Important Keywords:
try
except
else
finally
```

In [103]:

```
try:
    num1 = int(input("Enter the first number: "))
    num2 = int(input("Enter the second number: "))
    res = num1/num2
except ZeroDivisionError as z:
    print("Any value divided by zero gives error..\nto know more about error refer this
# except ValueError as v:
#     print("Error: ",v)
except Exception:
    print('Some Error ')
else:
    print("Result: ",res)
finally:
    print("Executed without any errors")
```

```
Enter the first number: a
Some Error
Executed without any errors
```

In [201]:

```
a=[40,80,160]
try:
    ind=int(input("enter the location"))
    a[ind]
except IndexError:
    print("Index not available")
except ValueError:
    print("Int value only allowed to access a element in a list")
else:
    print(a[ind])
finally:
    print("Done")
```

```
enter the locationaa
Int value only allowed to access a element in a list
Done
```

Module4

In [117]:

```
#(checking palindrome and excepting some errors )inside a class
class palindrome:
    try:
        num = int(input("Enter a Number: "))
        temp = num
        rev = 0
        while num>0:
            dig = num%10
            rev = rev*10+dig
            num = num//10
    except ValueError as v:
        print("Value Error: ",v)
    except ZeroDivisionError as z:
        print("Zero Division Error: ",z)
    except Exception as e:
        print("Error:",e)
    else:
        if temp == rev:
            print("Palindrome")
        else:
            print("Not a Palindrome")
    finally:
        print()
obj = palindrome()
```

```
Enter a Number: 15451
Palindrome
```

In [119]:

```
#single inheritance
class College:
    def show(self):
        print("This is a parent class named college")
class SRIHER(College):
    def display(self):
        print("This is a sub class of college")
obj = SRIHER()
obj.show()
obj.display()
```

```
This is a parent class named college
This is a sub class of college
```

In [120]:

```python
#multilevel inheritance:
class University:
    def show(self):
        print("This is a grandparent class named university")
class Sriher(University):
    def display(self):
        print("This is a parent class derived from University class")
class Sret(Sriher):
    def output(self):
        print("This is a child class derived from Sriher class and it can inherit featu
obj = Sret()
obj.show()
obj.display()
obj.output()
```

```
This is a grandparent class named university
This is a parent class derived from University class
This is a child class derived from Sriher class and it can inherit features
from University class and Sriher Class
```

In [121]:

```python
#multiple inheritance
class University:
    def show(self):
        print("This is a class named university")
class College:
    def display(self):
        print("This is a class named college")
class SRET(College,University):
    def output(self):
        print("This is a class inheriting feature from Univeristy and College class")
obj = SRET()
obj.show()
obj.display()
obj.output()
```

```
This is a class named university
This is a class named college
This is a class inheriting feature from Univeristy and College class
```

In [203]:

```python
#Hierarchical clustering
class University:
    def show(self):
        print("This is a class named university")
class College(University):
    def display(self):
        print("This is a class named college")
class SRET(University):
    def output(self):
        print("This is a class inheriting feature from Univeristy class")
#one parent class giving its features to its child class
#here college class and Sret class are sibling classess
obj1 = College()
obj2 = SRET()
obj1.show()
obj1.display()
obj2.show()
obj2.output()
```

```
This is a class named university
This is a class named college
This is a class named university
This is a class inheriting feature from Univeristy class
```

In [125]:

```python
#Polymorphism(Method Overriding)
class University:
    def show(self):
        print("This is a class named university")
class College(University):
    def show(self):
        super().show()
        print("This is a class named college")
obj = College()
obj.show()

```

```
This is a class named university
This is a class named college
```

In [204]:

```python
class College:
    def show(self,name):
        print("My name is: ",name)
class University(College):
    def show(self,name,age):
        print("My name is: ",name)
        print("My age is: ",age)
obj = University()
obj.show("Arjun",17)
obj1 = College()
obj1.show("Mike")
```

```
My name is:  Arjun
My age is:  17
My name is:  Mike
```

In [205]:

```python
#Polymorphism(Method Overloading)
class University:
    def show(self,name):
        print("My name is",name)
class college(University):
    def show(self,dept,age):
        print("My dept is",dept)
        print("My age is",age)
n = input("Enter name: ")
d = input("Enter dept: ")
a = int(input("Enter Age: "))
obj = college()
obj1 = University()
obj.show(d,a)
obj1.show(n)
```

```
Enter name: Arjun
Enter dept: CYS
Enter Age: 17
My dept is CYS
My age is 17
My name is Arjun
```

In [148]:

```
#bank acc
class Bank:
    def __init__ (self):
        self.balance = 0
        print("Your account is created")
    def deposit(self):
        amount = int(input("Enter the amount to be Deposited: "))
        self.balance+= amount
        #print("Your balance is: ",self.balance)
    def withdraw(self):
        amount = int(input("Enter the amount to be withdrawn: "))
        if amount<self.balance:
            self.balance-=amount
        else:
            print("Insufficient balance")
        #print("Your balance is: ",self.balance)

    def enquiry(self):
        print("Your balance is: ",self.balance)
ac = Bank()
while True:
    sel = int(input("1 for Deposit\n2 for Withdraw\n3 for Enquiry\n4 for exit"))
    if sel == 1:
        ac.deposit()
    elif sel == 2:
        ac.withdraw()
    elif sel == 3:
        ac.enquiry()
    else:
        break
```

```
Your account is created
1 for Deposit
2 for Withdraw
3 for Enquiry
4 for exit1
Enter the amount to be Deposited: 45
1 for Deposit
2 for Withdraw
3 for Enquiry
4 for exit4
```

Module5

```
regular expression:
match()-match only the first word
search()- search entire statement
sub()- replace
findall() - extract string based on pattern
```

In [151]:

```
import re
msg="Hello welcome to SRET"
p="Hello"
if re.match(p,msg):
    print("Found")
else:
    print("Not found")
```

```
Found
```

In [152]:

```
import re
msg="Hello welcome to SRET"
p="SRET"
if re.search(p,msg):
    print("Found")
else:
    print("Not found")
```

```
Found
```

In [153]:

```
# sub
msg="Welcome to SRET and SRET"
oldstring="SRET"
new="SRIHER"
msg1=re.sub(oldstring,new,msg,1)
print(msg1)
```

```
Welcome to SRIHER and SRET
```

In [154]:

```
# Extract date
msg = "Hey today date is 01-11-2022 http://google.com/11-11-2022"
srch = r'\d{2}-\d{2}-\d{4}'
a = re.findall(srch,msg)
print("Dates are:",a)
```

```
Dates are: ['01-11-2022', '11-11-2022']
```

In [155]:

```
# Extract Year
msg = "Hey today date is 01-11-2022 http://google.com/11-11-2022"
srch = r'\d{2}-\d{2}-(\d{4})'
a = re.findall(srch,msg)
print("Dates are:",a)
```

```
Dates are: ['2022', '2022']
```

In [156]:

```
#Extract the word starting with c and end with e
msg = "No clue is left"
p =r'\bc\S+e\b'
re.findall(p,msg)
```

Out[156]:

```
['clue']
```

In [157]:

```
#Write a program that seperates only the email id
p = r'[\w.]+@[\w.-]+'
msg = "Official mail id E022205.xx@gmail.com Private mail id dbbd@gmail.com"
a = re.findall(p,msg)
print("Valid Number: ",a)
```

```
Valid Number:  ['E022205.xx@gmail.com', 'dbbd@gmail.com']
```

In [158]:

```
#Write a program that validates a mobile phone number
#the number should strat with 7 or 8 or 9 followed by 9 digit
p = r'[7-9]{1}[0-9]{9}'
msg = "4578961234 7894561231 845786 9854761235"
a = re.findall(p,msg)
print("Valid Number: ",a)
```

```
Valid Number:  ['7894561231', '9854761235']
```

In [159]:

```
#"Python" = Py
import re
msg="Python"
p=r'\b\w\w'
re.findall(p,msg)
```

Out[159]:

```
['Py']
```

In [214]:

```
import re
msg = 7845784565255201
p = r'[3|6|8]{1}[0-9]{13}[[0]{1}[1-9]{1}]|[[1-9]{1}[0]{1}]'
re.findall(p,msg)
```

```
C:\Users\klmar\AppData\Local\Temp\ipykernel_3888\1100168740.py:4: FutureWarn
ing: Possible nested set at position 20
  re.findall(p,msg)
C:\Users\klmar\AppData\Local\Temp\ipykernel_3888\1100168740.py:4: FutureWarn
ing: Possible nested set at position 37
  re.findall(p,msg)
```

Out[214]:

```
[]
```

```
Create a triangle class in Python language, allowing you to build a triangle with length and breadth attributes. And perform the following operations a) Create a Perimeter() method to calculate the perimeter of the triangle and an Area() method to calculate the area of ••the triangle. b) Create a method display() that displays the length, breadth, perimeter, and area of an object created using an instantiation on the triangle class. c) Create a cuboid child class inheriting from the triangle class with a length, height and breadth attribute and another volume () method to calculate the volume of the cuboid.
```

In [213]:

```python
class Triangle:
    def __init__(self,l,b,s1,s2,s3):
        self.l = l
        self.b = b
        self.s1 = s1
        self.s2 = s2
        self.s3 = s3
    def Perimeter(self):
        print("The perimeter of the triangle is:",self.s1+self.s2+self.s3)
    def area(self):
        print("Area of the triangle is:",0.5*self.l*self.b)
    def display(self):
        print("The length of the triangle is ",self.l)
        print("The breadth of the triangle is ",self.b)
        print("The perimeter of the triangle is:",self.s1+self.s2+self.s3)
        print("Area of the triangle is:",0.5*self.l*self.b)
l = int(input("Enter the length: "))
w = int(input("Enter the width: "))
h = int(input("Enter the height: "))
class Cuboid(Triangle):
    def volume(self,a,b,c):
        print("VOlume of the cuboid is",a*b*c)

obj = Cuboid(5,6,7,8,9)
obj.Perimeter()
obj.area()
obj.display()
obj.volume(l,w,h)
```

```
Enter the length: 8
Enter the width: 2
Enter the height: 5
The perimeter of the triangle is: 24
Area of the triangle is: 15.0
The length of the triangle is  5
The breadth of the triangle is  6
The perimeter of the triangle is: 24
Area of the triangle is: 15.0
VOlume of the cuboid is 80
```

In [ ]:

```

```