MANAV RACHNA UNIVERSITY, FARIDABAD
Department of Computer Science and Technology

Course: B.Tech (CST)

Subject: Programming for Problem Solving using Python(CSW208B) Session: 2020-21

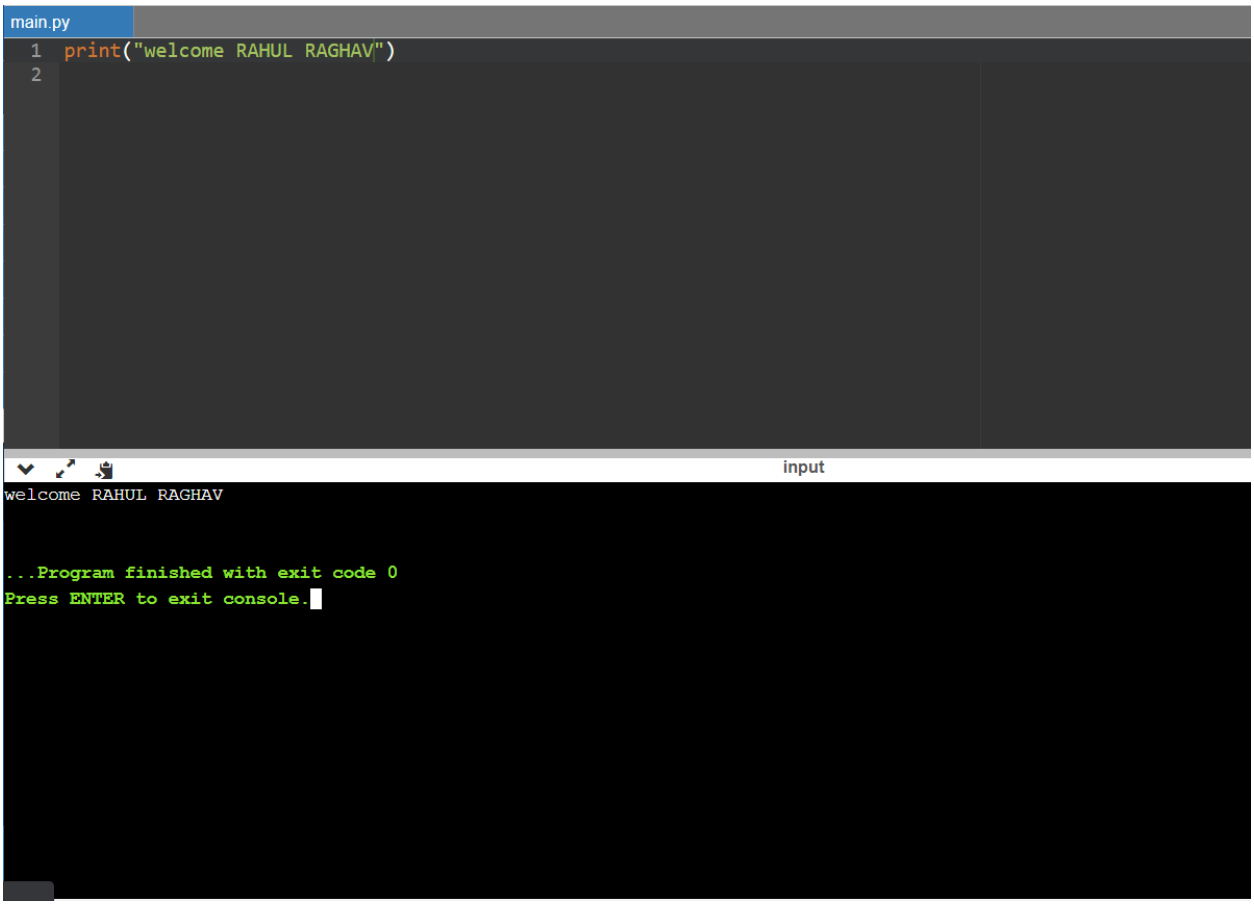
Lab 1: *Understanding Jupyter IDE for creating and executing a Python program*

Learning Outcome CO1**:** *Student will be able to get the Python environment up and running and the basics of Python programming language*

Blooms Taxonomy Level*: BT1, BT2*

1. Introducing the Python language, Understanding the Python shell.
2. Development environment setup, Configuring – Installation of Anaconda IDE.
3. Introduction to Jupyter notebook.
4. Working with Jupyter notebook.

5. Writing a program to print a welcome message.



6. Write a program to add two numbers.

CODE:

main.py

```
x = input("Type a number: ")
y = input("Type another number: ")

sum = int(x) + int(y)

print("The sum is: ", sum)
```

input

```
Type a number: 52
Type another number: 15
The sum is:  67


...Program finished with exit code 0
Press ENTER to exit console.
```

7. Write a program to take first name, middle name and last name from the user. Greet the user.

CODE:

```
fname=input("enter first name:")
mname=input("enter middle name:")
lname=input("enter last name :")

print("hello"," ",fname," ",mname," ",lname)
```

RAHUL RAGHAV **2K19CSUN01162** **CSE 4-C**

main.py

```python
fname=input("enter first name:")
mname=input("enter middle name:")
lname=input("enter last name :")

print("hello"," ",fname," ",mname," ",lname)

```

input

```
enter first name:rahul
enter middle name:
enter last name :raghav
hello   rahul      raghav


...Program finished with exit code 0
Press ENTER to exit console.
```

Lab -2: *Programming constructs in python -hands-on practice*

1. Check whether a number is even or odd

```python
num = int(input("Enter a number: "))
if (num % 2) == 0:
    print("{0} is Even".format(num))
else:
    print("{0} is Odd".format(num))
```

```
Enter a number: 43
is Odd
>
```

2. Check whether an entered year is leap year or not.

```python
year = 2021

if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".format(year))
        else:
            print("{0} is not a leap year".format(year))
    else:
        print("{0} is a leap year".format(year))
else:
    print("{0} is not a leap year".format(year))
```

```
2021 is not a leap year
>
```

3. Write a program to check whether a character is vowel or consonants.

main.py

```python
l = input("Input a letter of the alphabet: ")

if l in ('a', 'e', 'i', 'o', 'u'):
    print("%s is a vowel." % l)
elif l == 'y':
print("Sometimes letter y stand for vowel, sometimes stand for consonant.")
else:
print("%s is a consonant." % l)
```

Shell

```
Input a letter of the alphabet: a
is a vowel.
>
```

4. Write a program to find the smallest of two numbers.

main.py

```python
num1=int(input("Enter your first number:"))
num2=int(input("Enter your second number: "))
if(num1<num2):
    print("{} is smallest".format(num1))
elif(num2<num1):
    print("{} is smallest".format(num2))
else:
    print("{} and {} are equal".format(num1,num2))
```

Shell

```
Enter your first number:2
Enter your second number: 3
2 is smallest
>
```

5. Find the Factorial of a Number

main.py

```python
num = 7
factorial = 1

# check if the number is negative, positive or zero
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Shell

```
The factorial of 7 is 5040
>
```

6. Write a program to print this patterns

*

```
    *   *
  *   *   *
*   *   *   *
```

main.py

```python
n = int(input("Enter the number of rows: "))
m = (2 * n) - 2
for i in range(0, n):
    for j in range(0, m):
        print(end=" ")
    m = m - 1  # decrementing m after each loop
    for j in range(0, i + 1):
        # printing full Triangle pyramid using stars
        print("* ", end=' ')
    print(" ")
```

Output

```
Enter the number of rows: 4
       *
     *   *
   *   *   *
 *   *   *   *
```

7. Write a program to print this series

   1 1 2 3 5 8 13

main.py

```python
nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
elif nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1 + n2
        # update values
        n1 = n2
        n2 = nth
        count += 1
```

Shell

```
How many terms? 8
Fibonacci sequence:
0
1
1
2
3
5
8
13
>
```

8. Check whether a number is prime or not

main.py

```python
num = 2

# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2,num):
        if (num % i) == 0:
            print(num,"is not a prime number")
            break
    else:
        print(num,"is a prime number")

# if input number is less than
# or equal to 1, it is not prime
else:
    print(num,"is not a prime number")
```

Shell

```
2 is a prime number
>
```

9. Make a Simple Calculator.

main.py

```python
# This function adds two numbers
def add(x, y):
    return x + y

# This function subtracts two numbers
def subtract(x, y):
    return x - y

# This function multiplies two numbers
def multiply(x, y):
    return x * y

# This function divides two numbers
def divide(x, y):
    return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")
```

```python
while True:
    # Take input from the user
    choice = input("Enter choice(1/2/3/4): ")

    # Check if choice is one of the four options
    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))
        break
    else:
        print("Invalid Input")
```

Output

Shell Clear

```
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 3
Enter first number: 15
Enter second number: 14
15.0 * 14.0 = 210.0
>
```

RAHUL RAGHAV 2K19CSUN01162 CSE 4-C

LAB-3

1.WAPtodemonstratewhileloopwithelsestatement.

```
#whileloopwithelsestatement
i=int(input("Entertheintegerless10:"))whilei<10:print(i)
i+=1else:print("iisnolongerlessthan10")
```

```
Enter the integer less 10 : 5
5
6
7
8
9
i is no longer less than 10
```

2.Print1st5evennumbers(usebreakstatement).

```
#firstfiveprimenumbersusingbreakstatementi=0c=1
print("FirstfiveevennumbersEvennumbers")foriin
range(120):ifi%2==0: print(i) c+=1elifc>5: break
```

```
First five even numbersEven numbers
0
2
4
6
8
```

3.Print1st4evennumbers(usecontinuestatement).

```
#firstfourprimenumbersusingcontinuei=0c=1
print("FirstfiveevennumbersEvennumbers")foriin range(8):ifi%2!=0:
c+=1 continueelifc<5: print(i)
```

```
First five even numbersEven numbers
0
2
4
6
```

4.WAPtodemonstratePassstatements.

```
#Demonstratingpassstatementa=
10b=20if(a<b):passelse: print("b<a")
```

```
a = 10
b = 20
if(a<b):
    pass
else:
    print("b<a")
```

5.WriteaPythonprogramtocalculatethelengthofastring.

```
#Tocalculatethelengthofagivenstrings=input("Enter
thestring:")print("Lengthofthestringis",len(s))
```

```
enter a string : python workshop
Length of string is 15
```

6.WriteaPythonprogramtocountthenumberofcharacters (characterfrequency)inastring

```
string=input()f={} foriinstring:
    f[i]=f.get(i,0)+1print(f)
```

```
python noob
{'p': 1, 'y': 1, 't': 1, 'h': 1, 'o': 3, 'n': 2, ' ': 1, 'b': 1}
```

7.WriteaPythonprogramtogetastringmadeofthefirst2 andthelast2charsfromagivenastring.Ifthestringlength islessthan2,returninsteadoftheemptystring.

```
#Togetastringmadeofthefirst2andthelast2charsfromagivenastring
s=input("Enterthestring:")print("Requiredoutput:
"+s[0:2]+s[-2:])
```

```
Enter the string: PYTHON NOOB
Required output: PYOB
```

8.WriteaPythonprogramtogetasinglestringfromtwogiven strings,separatedbyaspaceandswapthefirsttwo charactersofeachstring.

```
a=input()b=input()x=a[0:2]
a=a.replace(a[0:2],b[0:2])
b=b.replace(b[0:2],x)print(a,b)
```

```
python
noob
nothon pyob
```

9.WriteaPythonprogramtoadd'ing'attheendofagiven string (lengthshouldbeatleast3).Ifthegivenstringalreadyends with'ing'thenadd'ly'instead.Ifthestringlengthofthe givenstringislessthan3,leaveitunchanged

```
s=input()iflen(s)<3: print(s)elifs[-3:]=='ing':
        print(s+'ly')else:print(s
                                +'ing')
```

```
hypothetically
hypotheticallying
```

LAB-4

Tuple

1. Write a Python program to create a tuple.

In [1]:
```
# Empty tuple
my_tuple = ()
print(my_tuple)

# Tuple having integers
my_tuple = (1, 2, 3)
print(my_tuple)

# tuple with mixed datatypes
my_tuple = (1, "Hello", 3.4)
print(my_tuple)

# nested tuple
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
```

```
()
(1, 2, 3)
(1, 'Hello', 3.4)
('mouse', [8, 4, 6], (1, 2, 3))
```

2. Write a Python program to create a tuple with different data types.

In [2]:
```
tuplex = ("tuple", False, 3.2, 1)
print(tuplex)
```

```
('tuple', False, 3.2, 1)
```

3. Write a Python program to create a tuple with numbers and print one item.

In [3]:
```
tuplex = 5, 10, 15, 20, 25
print(tuplex)

tuplex = 5,
print(tuplex)
```

```
(5, 10, 15, 20, 25)
(5,)
```

4. Write a Python program to unpack a tuple in several variables.

```
In [4]: tuplex = 4, 8, 3
        print(tuplex)
        n1, n2, n3 = tuplex
        # unpack a tuple in variables
        print(n1 + n2 + n3)
        # the number of variables must be equal to the number of items of the tuple
        n1, n2, n3, n4 = tuplex
```

```
(4, 8, 3)
15

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-2c2162337212> in <module>
      5 print(n1 + n2 + n3)
      6 # the number of variables must be equal to the number of items of the tuple
----> 7 n1, n2, n3, n4 = tuplex

ValueError: not enough values to unpack (expected 4, got 3)
```

5. Write a Python program to add an item in a tuple.

```
In [5]: tuplex = (4, 6, 2, 8, 3, 1)
        print(tuplex)

        tuplex = tuplex + (9,)
        print(tuplex)

        tuplex = tuplex[:5] + (15, 20, 25) + tuplex[:5]
        print(tuplex)
        listx = list(tuplex)
        listx.append(30)
        tuplex = tuple(listx)
        print(tuplex)
```

```
(4, 6, 2, 8, 3, 1)
(4, 6, 2, 8, 3, 1, 9)
(4, 6, 2, 8, 3, 15, 20, 25, 4, 6, 2, 8, 3)
(4, 6, 2, 8, 3, 15, 20, 25, 4, 6, 2, 8, 3, 30)
```

6. Write a Python program to convert a tuple to a string.

```
In [6]: tup = ('e', 'x', 'e', 'r', 'c', 'i', 's', 'e', 's')
        str =  ''.join(tup)
        print(str)
```

```
exercises
```

7. Write a Python program to get the 4th element and 4th element from last of a tuple

```
In [5]: tuplex = ("P", 1, "t", "H", "O", "N", "P", "R", "O")
        print(tuplex)
        item = tuplex[3]
        print(item)
        item1 = tuplex[-4]
        print(item1)

        ('P', 1, 't', 'H', 'O', 'N', 'P', 'R', 'O')
        H
        N
```

8. Write a Python program to create the colon of a tuple.

```
In [9]: from copy import deepcopy
        tuplex = ("HELLO", 5, [], True)
        print(tuplex)
        tuplex_colon = deepcopy(tuplex)
        tuplex_colon[2].append(50)
        print(tuplex_colon)
        print(tuplex)

        ('HELLO', 5, [], True)
        ('HELLO', 5, [50], True)
        ('HELLO', 5, [], True)
```

9. Write a Python program to find the repeated items of a tuple.

```
In [10]: tuplex = 2, 4, 5, 6, 2, 3, 4, 4, 7
         print(tuplex)
         count = tuplex.count(4)
         print(count)

         (2, 4, 5, 6, 2, 3, 4, 4, 7)
         3
```

10. Write a Python program to check whether an element exists within a tuple.

```
In [11]: tuplex = ("w", 3, "r", "e", "s", "o", "u", "r", "c", "e")
         print("r" in tuplex)
         print(5 in tuplex)

         True
         False
```

11. Write a Python program to convert a list to a tuple.

```
In [12]: listx = [5, 10, 7, 4, 15, 3]
         print(listx)

         tuplex = tuple(listx)
         print(tuplex)

         [5, 10, 7, 4, 15, 3]
         (5, 10, 7, 4, 15, 3)
```

12. Write a Python program to remove an item from a tuple.

```
In [4]: tuplex = "H", 3, "L", "L", "O", "F", "R", "I", "N","D"
        print(tuplex)
        tuplex = tuplex[:2] + tuplex[3:]
        print(tuplex)
        listx = list(tuplex)

        listx.remove("N")
        tuplex = tuple(listx)
        print(tuplex)

        ('H', 3, 'L', 'L', 'O', 'F', 'R', 'I', 'N', 'D')
        ('H', 3, 'L', 'O', 'F', 'R', 'I', 'N', 'D')
        ('H', 3, 'L', 'O', 'F', 'R', 'I', 'D')
```

13. Write a Python program to slice a tuple

```
In [14]: tuplex = (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)
         _slice = tuplex[3:5]
         print(_slice)
         _slice = tuplex[:6]
         print(_slice)
         _slice = tuplex[5:]
         print(_slice)
         _slice = tuplex[:]
         print(_slice)
         _slice = tuplex[-8:-4]
         print(_slice)
         tuplex = tuple("HELLO WORLD")
         print(tuplex)
         _slice = tuplex[2:9:2]
         print(_slice)
         _slice = tuplex[::4]
         print(_slice)
         _slice = tuplex[9:2:-4]
         print(_slice)
```

```
(5, 4)
(2, 4, 3, 5, 4, 6)
(6, 7, 8, 6, 1)
(2, 4, 3, 5, 4, 6, 7, 8, 6, 1)
(3, 5, 4, 6)
('H', 'E', 'L', 'L', 'O', ' ', 'W', 'O', 'R', 'L', 'D')
('L', 'O', 'W', 'R')
('H', 'O', 'R')
('L', ' ')
```

14. Write a Python program to find the index of an item of a tuple.

```
In [15]: tuplex = tuple("index tuple")
         print(tuplex)
         index = tuplex.index("p")
         print(index)
         index = tuplex.index("p", 5)
         print(index)
         index = tuplex.index("e", 3, 6)
         print(index)
         index = tuplex.index("y")

         ('i', 'n', 'd', 'e', 'x', ' ', 't', 'u', 'p', 'l', 'e')
         8
         8
         3

         ---------------------------------------------------------------------------
         ValueError                                Traceback (most recent call last)
         <ipython-input-15-7f06c6a26ab5> in <module>
               7 index = tuplex.index("e", 3, 6)
               8 print(index)
         ----> 9 index = tuplex.index("y")

         ValueError: tuple.index(x): x not in tuple
```

15. Write a Python program to find the length of a tuple.

```
In [2]: tuplex = tuple("PYTHON_PROGRAMMING")
        print(tuplex)
        print(len(tuplex))

        ('P', 'Y', 'T', 'H', 'O', 'N', '_', 'P', 'R', 'O', 'G', 'R', 'A', 'M', 'M', 'I', 'N', 'G')
        18
```

16. Write a Python program to reverse a tuple.

```
In [1]: x = ("HELLO,FRIENDS")
        y = reversed(x)
        print(tuple(y))
        x = (5, 10, 15, 20)
        y = reversed(x)
        print(tuple(y))

        ('S', 'D', 'N', 'E', 'I', 'R', 'F', ',', 'O', 'L', 'L', 'E', 'H')
        (20, 15, 10, 5)
```

Lab -5

List

1. Write a Python program to sum all the items in a list.

In [1]:
```
# Python program to find sum of elements in list
total = 0

# creating a list
list1 = [11, 5, 17, 18, 23]

# Iterate each element in list
# and add them in variale total
for ele in range(0, len(list1)):
    total = total + list1[ele]

# printing total value
print("Sum of all elements in given list: ", total)
```

```
Sum of all elements in given list:  74
```

2. Write a Python program to multiplies all the items in a list.

In [2]:
```
def multiply_list(items):
    tot = 1
    for x in items:
        tot *= x
    return tot


print(multiply_list([1, 2, -8]))
```

```
-16
```

3. Write a Python program to get the largest number from a list

In [3]:
```
# Python program to find largest
# number in a list

# list of numbers
list1 = [10, 20, 4, 45, 99]

# sorting the list
list1.sort()

# printing the last element
print("Largest element is:", list1[-1])
```

```
Largest element is: 99
```

4. Write a Python program to get the smallest number from a list.

```
In [4]: # Python program to find smallest
        # number in a list

        # list of numbers
        list1 = [10, 20, 1, 45, 99]

        # printing the maximum element
        print("Smallest element is:", min(list1))

        Smallest element is: 1
```

5. Write a Python 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

```
In [5]: def match_words(words):
            ctr = 0

            for word in words:
                if len(word) > 1 and word[0] == word[-1]:
                    ctr += 1
            return ctr


        print(match_words(['abc', 'xyz', 'aba', '1221']))

        2
```

6. 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.

```
In [6]: def last(n): return n[-1]


        def sort_list_last(tuples):
            return sorted(tuples, key=last)


        print(sort_list_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))

        [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]
```

7. Write a Python program to remove duplicates from a list.

```
In [7]: a = [10,20,30,20,10,50,60,40,80,50,40]

        dup_items = set()
        uniq_items = []
        for x in a:
            if x not in dup_items:
                uniq_items.append(x)
                dup_items.add(x)

        print(dup_items)

        {40, 10, 80, 50, 20, 60, 30}
```

8. Write a Python program to check a list is empty or not.

```
In [8]: l = []
        if not l:
          print("List is empty")

        List is empty
```

9. Write a Python program to clone or copy a list.

```
In [9]: original_list = [10, 22, 44, 23, 4]
        new_list = list(original_list)
        print(original_list)
        print(new_list)

        [10, 22, 44, 23, 4]
        [10, 22, 44, 23, 4]
```

10. Write a Python program to find the list of words that are longer than n from a given list of words.

```
In [10]: def long_words(n, str):
             word_len = []
             txt = str.split(" ")
             for x in txt:
                 if len(x) > n:
                     word_len.append(x)
             return word_len


         print(long_words(3, "The quick brown fox jumps over the lazy dog"))

         ['quick', 'brown', 'jumps', 'over', 'lazy']
```

11. Write a Python function that takes two lists and returns True if they have at least one common member.

```
In [11]: def common_data(list1, list2):
             result = False
             for x in list1:
                 for y in list2:
                     if x == y:
                         result = True
                         return result


         print(common_data([1, 2, 3, 4, 5], [5, 6, 7, 8, 9]))
         print(common_data([1, 2, 3, 4, 5], [6, 7, 8, 9]))

         True
         None
```

12. Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

```
In [12]: color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
         color = [x for (i,x) in enumerate(color) if i not in (0,4,5)]
         print(color)

         ['Green', 'White', 'Black']
```