## Activity 2

**Aim:** Write a python program to print given string using indentation (space between characters).

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| def print\_with\_indentation(string):  indented\_string = ""  for i in string:  indented\_string += " " + i + " "  print(indented\_string)  # Example usage:  my\_string = "Hello, World!"  print\_with\_indentation(my\_string) |

## Output:

|  |
| --- |
|  |

## Activity 3

**Aim:** Define Integer Variables, floating variables and string variables.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| a = 10  name = "Padam Singh"  print('Int',int(a))  print('Float',float(a))  print(type(name)) |

## Output:

|  |
| --- |
|  |

## Activity 4

**Aim:** Write a program to add numbers and strings to the correct list using the append list method.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| numbers = [1,2,3]  strings = ["Hello","World"] # declare and initialize list  names = ["John", "Eric", "Jessica"]  second\_name = names[1]  print(numbers) # print the values of list  print(strings)  print("The second name on the names list is", second\_name) |

## Output:

|  |
| --- |
|  |

## Activity 5

**Aim:** Write a python program to add, subtract, multiply and divide given two numbers by using arithmetic operators.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| num1 = int(input('Enter the first number: '))  num2 = int(input('Enter the second number: '))  Addition = num1 + num2  Subtraction = num1 - num2  Division = num1 / num2  Modulus = num1 % num2  FloorDivision = num1 // num2  Multiplication = num1 \* num2  Exponentiation = num1 \*\* num2  print('Addition' ,Addition)  print('Subtraction' ,Subtraction)  print('Division' ,Division)  print('Modulus' ,Modulus)  print('Floor Division',FloorDivision)  print('Multiplication ',Multiplication)  print('Exponentiation ',Exponentiation ) |

## Output:

|  |
| --- |
|  |

## Activity 6

**Aim:** Write a python program multiplying strings to form a string with repeating sequence.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| x = 'Welcome '  y = 'python '  print ((x+y)\*5) |

## Output:

|  |
| --- |
|  |

## Activity 7

**Aim:** Write a Python program to get the largest number from a list by using max and min commands.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_list = [1, 2, 3]  print('Max value in list',max(my\_list)) # max() finds the maximum among values  print('Min value in list',min(my\_list)) # min() finds the minimum among values |

## Output:

|  |
| --- |
|  |

## Activity 8

**Aim:** Write a Python program to find whether a given number (accept from the user) is even or odd by using if else command.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| num = int(input('Enter a Number: '))  if num%2==0:  print(f'{num} is Even')  else:  print(f'{num} is Odd') |

## Output:

|  |
| --- |
|  |

## Activity 9

**Aim:** Write a Python program to create a histogram from a given list of integers by using for while loop.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 1 hour

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| def histogram( items ): # create function  for n in items: # for loop through values of n  output = ''  times = n  while( times > 0 ): # check condition of counter  output += '\*'  times = times - 1 # decrement the counter  print(output) # print the output  histogram([2, 3, 6, 5]) |

## Output:

|  |
| --- |
|  |

## Activity 10

**Aim:** Write a Python program to compute the greatest common divisor (GCD) of two positive integers by using loop.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| #**Example 1**  def gcd(x, y): # create function gcd()  gcd = 1  if x % y == 0:  return y  for k in range(int(y / 2), 0, -1): # for loop from y/2 to 0 increment -1  if x % k == 0 and y % k == 0:  gcd = k  break  return gcd  print(gcd(12, 17)) # print the gcd value  print(gcd(4, 6))  #**Example 2**  a=int(input("first value:"))  new\_a=[]  for i in range(1,a+1):  if a%i==0:  new\_a.append(i)  b=int(input("second value:"))  new\_b=[]  for k in range(1,b+1):  if b%k==0:  new\_b.append(k)  same= set(new\_a) & set(new\_b)  gcd= max(same)  print(a,"Divisable by",new\_a)  print(b,"Divisable by",new\_b)  print("Greatest Comman Divisor",gcd) |

## Output:

|  |
| --- |
|  |

## Activity 11

**Aim:** Write a Python program to get the least common multiple (LCM) of two positive integers using if else and while commands.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| #Example 1  def lcm(x,y):  if x>y:  z=x  else:  z=y  while(True):  if((z%x==0) and (z%y==0)):  lcm=z  break  z+=1  return (f"Lcm of {x} and {y} is : {lcm}")  x = int(input("Insert 1st value :"))  y = int(input("Insert 2nd value :"))  lcm(x,y)  #Example 2  a=int(input("first value:"))  new\_a=[]  for i in range(1,a+1):  if a%i==0:  new\_a.append(i)  b=int(input("second value:"))  new\_b=[]  for k in range(1,b+1):  if b%k==0:  new\_b.append(k)  same= set(new\_a) & set(new\_b)  gcd= max(same)  multi = a\*b  lcm = multi/gcd  print(a,"Divisable by",new\_a)  print(b,"Divisable by",new\_b)  print("Greatest Comman Divisor",lcm) |

## Output:

|  |
| --- |
|  |

## Activity 12

**Aim:** Write a Python program to sort (ascending and descending) a dictionary by value.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_dict = {1:2, 3:4, 4:3, 2:1, 0:0}  item = sorted(my\_dict.items())  print(f"Dictionary Ascending order by value {item}")  item.reverse()  print(f"Dictionary Descending order by value {item}") |

## Output:

|  |
| --- |
|  |

## Activity 13

**Aim:** Write a Python program to create a tuple.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_tuple = (1,2,3, 'a', 'b', 'True', True, ['apple','banana'], ('one', 'two'))  my\_tuple |

## Output:

|  |
| --- |
|  |

## Activity 14

**Aim:** Write a Python program to create a tuple with different data types.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_tuple = (1,2,3, 'a', 'b', 'True', True)  my\_tuple |

## Output:

|  |
| --- |
|  |

## Activity 15

**Aim:** Write a Python program to create a set.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_set = set()  print(my\_set)  my\_set1 = set([1,2,3])  print(type(my\_set1)) |

## Output:

|  |
| --- |
|  |

## Activity 16

**Aim:** Write a Python program to add member(s) in a set.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_set = set()  print(my\_set)  my\_set.add('Red')  my\_set |

## Output:

|  |
| --- |
|  |

## Activity 17

**Aim:** Write a Python program to find maximum and the minimum value in a set.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_set = set([1,2,3,5,6,7,8,9,10])  print(f"Maximum value of set: {min(my\_set)}")  print(f"hello {my\_set}")  print(f"Minimum value of set: {max(my\_set)}") |

## Output:

|  |
| --- |
|  |

## Activity 18

**Aim:** Write a Python program to find the length of a set.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| my\_set = {'one', 'two', 3, 4, True, None}  len(my\_set) |

## Output:

|  |
| --- |
|  |

## Activity 19

**Aim:** Write a Python program to convert temperatures to and from Centigrade to Fahrenheit.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| Fahrenheit = float(input("Enter a temperature in Fahrenheit: "))  Celsius = (Fahrenheit - 32) \* 5.0/9.0  print ("Temperature:", Fahrenheit, "Fahrenheit = ", Celsius, " C")  Celsius = float(input("Enter a temperature in Celsius: "))  Fahrenheit = 9.0/5.0 \* Celsius + 32  print ("Temperature:", Celsius, "Celsius = ", Fahrenheit, " F") |

## Output:

|  |
| --- |
|  |

## Activity 20

**Aim:** Write a python program to find Fibonacci series.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| n = int(input("How many terms?: "))  x=0  y=1  z=0  for i in range(0,n,1):  print(z ,end=" ")  x=y  y=z  z=x+y |

## Output:

|  |
| --- |
|  |

## Activity 21

**Aim:** Write a python program to find factorial using function in Python IDLE.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| n = int(input('Insert You Number: '))  def fc(n):  factorial = 1  for i in range(1, n+1 ):  factorial \*= i  return factorial    result = fc(n)  print("The factorial of", n, "is", result) |

## Output:

|  |
| --- |
|  |

## Activity 22

**Aim:** Write a python program to find whether the given string is palindrome or not by using function.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| def palindrome(n):  if n==n[::-1]:  print(f"Given value {n} is palindrome.")  else:  print(f"Given value {n} is not palindrome.")  palindrome("mom")  palindrome("dad")  palindrome("meggm") |

## Output:

|  |
| --- |
|  |

## Activity 23

**Aim:** Write a python class to reverse a string word by word.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| word = str(input("Enter you word for reverse"))  print(word[::-1]) |

## Output:

|  |
| --- |
|  |

## Activity 24

**Aim:** Write a python class named as circle by a radius and two methods of computer area and perimeter of a circle.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| import math  class circle():  def \_\_init\_\_(self,radius):  self.radius=radius  def area(self):  return math.pi\*(self.radius\*\*2)  def perimeter(self):  return 2\*math.pi\*self.radius    value\_of\_circle= int(input("Enter radius of circle: "))  obj=circle(value\_of\_circle)  print(round(obj.area(),2))  print(round(obj.perimeter())) |

## Output:

|  |
| --- |
|  |

## Activity 25

**Aim:** Write a python program to sort a list of elements using bubble sort algorithm.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
| #Example 1  def bubblesort(my\_list):  n = len(my\_list)    for i in range(n-1):  for j in range(0, n-i-1):    if my\_list[j] > my\_list[j+1]:  swapped = True  my\_list[j], my\_list[j+1] = my\_list[j+1], my\_list[j]  if not swapped:  return    my\_list = [5,1,6,4,8,7,9,3,2,10]  bubblesort(my\_list)  print("Sorted array is :")  for i in range(len(my\_list)):  print(my\_list[i], end= " ")  #Example 2  def bubblesort(n):  for i in range(len(n)-1,0,-1):  for j in range(i):  if n[j]>n[j+1]:  n[j],n[j+1]=n[j+1],n[j]    n=[5,1,6,4,8,7,9,3,2,10]  bubblesort(n)  print(n) |

## Output:

|  |
| --- |
| Example 1    Example 2 |

## Activity 26

**Aim:** Write a python program to copy content of a file to another file.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

We need to create 2 .txt file into the same location where you are saving your jupyter notebook file.

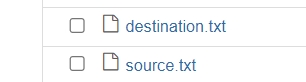
I am saving my all file inside the Core Module 4 folder (give your folder name where you are saving your file)



Step 1. Create a source.txt file and add below line.

|  |
| --- |
| Hello, This is source.txt file. |

Step 2. Create destination.txt file and leave it blank.



## Input:

|  |
| --- |
| with open("source.txt", 'r') as source:  file\_content = source.read()  with open("destination.txt", 'w') as destination:  destination.write(file\_content)  print("Your file content successfuly") |

## Output:

|  |
| --- |
| Code output:    destination.txt file output after code. |

## Activity 27

**Aim:** Write a python program to find the frequency of words in a file.

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

We need to create a txt file where you are save your jupyter notebook text.txt and write some line into this.

## Input:

|  |
| --- |
| from collections import Counter  # Read the file  with open('text.txt', 'r') as file:  text = file.read()  # Tokenize the text into words and count their frequencies  word\_freq = Counter(text.split())  # Print the word frequencies  for word, freq in word\_freq.items():  print(f'{word}: {freq}') |

## Output:

|  |
| --- |
|  |

## Activity 28

**Aim:** .

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
|  |

## Output:

|  |
| --- |
|  |

## Activity 29

**Aim:** .

# **Learning outcome**: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

###### Duration: 2 hours

# List of Hardware/Software requirements:

* Laptop/Computer with windows
* Python Software/Jupyter notebook

# Program:

## Input:

|  |
| --- |
|  |

## Output:

|  |
| --- |
|  |