PYTHON

Assignment no.1

CT University,Ferozpur Road Ludhiana

- Submitted by: CHAHAT
- Registration No.: 72513852
- Programme: BCA(AI&DS)
- Submitted to:NIKHIL PANDEY SIR

1.Write a python programme that prints “Hello,World!”.

INPUT:

```
print("hellO World")
```

OUTPUT:

```
======================
hello World
```

2. Take the user's name as input and print "Hello,<name>!".

INPUT:

```
name= input("enter your name:")
print("Hello,"+name+"!")
```

OUTPUT:

```
enter your name:CHAHAT
Hello,CHAHAT!
```

3.Ask the user for their age and print how old they will be in 5.

INPUT:

```
age = int(input("Enter your age:"))
future_age = age + 5
print(" you will be", future_age,"years old.")
```

OUTPUT:

```
==============================
Enter your age:18
 you will be 23 years old.
```

4. Take two numbers as input and print their sum.

INPUT:

```
num1 = float(input("Enter the first number:"))
num2 = float(input("Enter the second number:"))
sum = num1 + num2
print("the sum of", num1 ,"and",num2,"is",sum)
```

OUTPUT:

```
================================
Enter the first number:7
Enter the second number:8
the sum of 7.0 and 8.0 is 15.0
```

5. Ask the user for a number and print whether it's even or odd

INPUT:

```
num = int(input("Enter a number:"))
if num % 2 == 0:
    print(num,"is even")
else:
    print(num,"is odd")
```

OUTPUT:

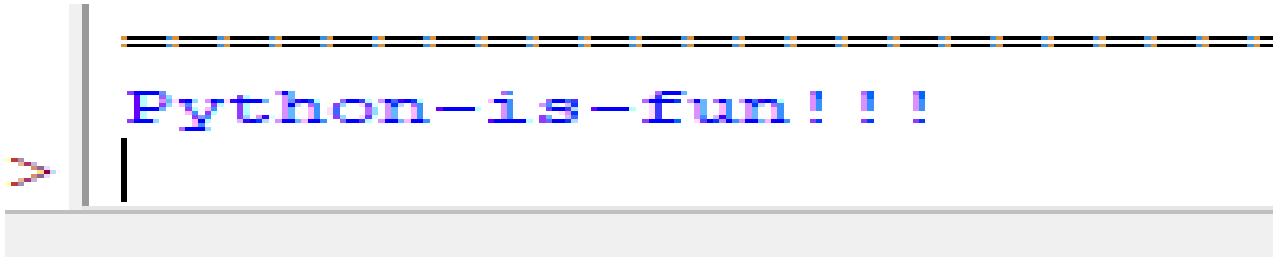
```
==========================
Enter a number:8
8 is even
```

6. Use the sep and end parameters in print() to print “Python is fun” on one line.

INPUT:

```
print("Python","is","fun",sep="-",end="!!!")
```

OUTPUT:



7.Write a program that takes the name and favourite color of a user and prints:”Alice likes blue”.

INPUT:

```
name = input("Enter your name:")
color = input("Enter your favourite color:")
print(name,"likes",color)
```

OUTPUT:

```
--------------------------------------
Enter your name:alice
Enter your favourite color:blue
alice likes blue
>
```

8.Use input() to take the length and width of a rectangle and print its area.

INPUT:

```
length = float(input("Enter the length of the rectangle:"))
width = float(input("Enter the width of the rectangle:"))
area = length * width
print("the area of the rectangle is",area)
```

OUTPUT:

```
Enter the length of the rectangle:8
Enter the width of the rectangle:10
the area of the rectangle is 80.0
```

9. Accept a number using input() and check whether it is greater than 100.

INPUT:

```
num = float(input("Enter a number:"))
if num > 100:
    print(num,"is greater than 100")
else:
    print(num,"is not greater than 100.")
```

OUTPUT:

```
Enter a number:5
5.0 is not greater than 100.
```

10. Use input() to get three numbers and print their average.

INPUT:

```
num1 = float(input("Enter the first number:"))
num2 = float(input("Enter the second number:"))
num3 = float(input("Enter the third number:"))
average = (num1 + num2 + num3)/3
print("The average of the three numbers is",average)
```

OUTPUT:

```
========== RESTART:
Enter the first number:9
Enter the second number:8
Enter the third number:6
The average of the three numbers is 7.666666666666667
>
```

11.Accept an integer from the user and print its square.

INPUT:

```
num = int(input("Enter an integer:"))
square = num**2
print("the square of",num,"is",square)
```

OUTPUT:

```
==============================
Enter an integer:55
the square of 55 is 3025
>
```

12.Accept a float from the user and round it to 2 decimal places.

INPUT:

```
num = float(input("Enter a float number:"))
rounded_num = round(num,2)
print("The number rounded to 2 decimal place is",rounded_num)
```

OUTPUT:

```
==========
Enter a float number:45
The number rounded to 2 decimal place is 45.0
```

13.Add two float numbers and print the result.

INPUT:

```
num = float(input("Enter a float number:"))
rounded_num = round(num,2)
PRINT("The number rounded to 2 decimal places is",rounded_num)
```

OUTPUT:

```
----------------------------------------------------

Enter a float number:45
The number rounded to 2 decimal place is 45.0
>

====================================================
```

14.Multiply a float and an integer and print the result.

INPUT:

```
float_num = float(input("Enter a float number:"))
int_num = int(input("Enter an integer:"))
result = float_num*int_num
print("The result of multiplying",float_num,"and",int_num,"is",result)
```

OUTPUT:

```
================================================================================
Enter a float number:45
Enter an integer:7
The result of multiplying 45.0 and 7 is 315.0
```

15. Accept a complex and print its real and imaginary parts.

INPUT:

```
num = complex(input("Enter a complex number(e.g., 3+4j):"))
print("The real part is ",num.real)
print("The imaginary part is",num.imag)
```

OUTPUT:

```
================================================================
Enter a complex number(e.g., 3+4j):5+7j
The real part is  5.0
The imaginary part is 7.0
>
```

16. Convert a float into an int and print both values.

INPUT:

```
num = float(input("Enter afloat number:"))
num_int = int(num)
print("Original float value:",num)
print("Converted integer value:",num_int)
```

OUTPUT:

```
==========================================
Enter afloat number:29.5
Original float value: 29.5
Converted integer value: 29
>
```

17.Convert an int to a float and print both values.

INPUT:

```
num = int(input("Enter an integer:"))
num_float = float(num)
print("Original integer value:",num)
print("Converted float value:",num_float)
```

OUTPUT:

```
--------------------------------------------------
Enter an integer:42
Original integer value: 42
Converted float value: 42.0
>
```

18.Accept two numbers(one int,one float), and print the result of int + float.

INPUT:

```
float_num = float(input("Enter a float number:"))
int_num = int(input("Enter an integer:"))
result = float_num+int_num
print("The sum of",float_num,"and",int_num,"is",result)
```

OUTPUT:

```
Enter a float number:24.5
Enter an integer:26
The sum of 24.5 and 26 is 50.5
```

19. Create a complex number using complex(a,b) and print it.

INPUT:

```
a = float(input("Enter the real part:"))
b = float(input("Enter the imaginary part:"))
num = complex(a,b)
print("The complex number is",num)
```

OUTPUT:

```
==========================================
Enter the real part:26
Enter the imaginary part:24
The complex number is (26+24j)
>
```

20. Add two complex numbers and print the result.

INPUT:

```
a1 = float(input("Enter the real part of the first complex number:"))
b1 = float(input("Enter the imaginary part of the first complex number:"))
a2 = float(input("Enter the real part of the second complex number:"))
b2 = float(input("Enter the imaginary part of the second complex number:"))
num1 = complex(a1,b1)
num2 = complex(a2,b2)
result = num1 = num2
print("The sum of",num1,"and",num2,"is",result)
```

OUTPUT:

```
Enter the real part of the first complex number:3
Enter the imaginary part of the first complex number:4
Enter the real part of the second complex number:5
Enter the imaginary part of the second complex number:6
The sum of (5+6j) and (5+6j) is (5+6j)
```

21.Use arithmetic operators to calculate (10+5)*3-8/2.

INPUT:

```
result = (10+5)*3-8/2
print("The result of (10+5)*3-8/2 is",result)
```

OUTPUT:

```
==========================================
The result of (10+5)*3-8/2 is 41.0
```

22.Compare two integers using all comparison operators.

INPUT:

```
num1 = int(input("Enter the first integer:"))
num2 = int(input("Enter the second integer:"))
print(f"{num1} == {num2} : {num1 == num2}")
print(f"{num1} !+ {num2} : {num1 != num2}")
print(f"{num1} >  {num2} : {num1 >  num2}")
print(f"{num1} <  {num2} : {num1 <  num2}")
print(f"{num1} >= {num2} : {num1 >= num2}")
print(f"{num1} <= {num2} : {num1 <= num2}")
```

OUTPUT:

```
================================
Enter the first integer:24
Enter the second integer:28
24 == 28 : False
24 !+ 28 : True
24 >  28 : False
24 <  28 : True
24 >= 28 : False
24 <= 28 : True
```

23.Use logical operators to check if a number is between 10 and 20.

INPUT:

```
num = float(input("Enter a number:"))
if num > 10 and num < 20:
    print(f"{num}is between 10 and 20.")
else:
    print(f"{num} is not between 10 and 20.")
```

OUTPUT:

```
--------------------------------------
Enter a number:29
29.0 is not between 10 and 20.
```

24.Use and,or and not in a single expression.

INPUT:

```
num = float(input("Enter a number:"))
if (num>0 and num < 10) or not (num == 5):
    print(f"The expression is true for{num}")
else:
    print(f"the expression is false for {num}")
```

OUTPUT:

```
Enter a number:27
The expression is true for27.0
```

25.Use bitwise&,|on two integers and print the result.

INPUT:

```
num1 = int(input("Enter the first integer:"))
num2 = int(input("Enter the second ineger:"))

and_result = num1 & num2
or_result = num1 | num2

print(f"{num1} & {num2} = {and_result}")
print(f"{num1} | {num2} = {or_result}")
```

OUTPUT:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2
Enter "help" below or click "Help" above for
>>>
================== RESTART: C:/Users/acer/Docu
Enter the first integer:26
Enter the second ineger:24
26 & 24 = 24
26 | 24 = 26
>>>
```

26. Use the assignment operators +=,-=,*=,/= and show results.

INPUT:

```
num = 10
print(f"Initial value: {num}")
num += 5
print(f"After += 5: {num}")
num -= 3
print(f"After -= 3: {num}")
num *= 2
print(f"After *= 2: {num}")
num /= 4
print(f"After /= 4:{num}")
```

OUTPUT:

```
==========
Initial value: 10
After += 5: 15
After -= 3: 12
After *= 2: 24
After /= 4:6.0
```

27.Check if two variables point to the same object using is and is not.

INPUT:

```
a = [1, 2, 3]
b = a
c = [1, 2, 3]
print(f"a is b: {a is b}")
print(f"a is c: {a is c}")
print(f"a is not c: { a is not c}")
print(f"a is not b: { a is not b}")
```

OUTPUT:

```
================================
a is b: True
a is c: False
a is not c: True
a is not b: False
>>
```

28. Check if a character exists in a string using in and not in.

INPUT:

```
text = input("Enter a string:")
char = input("Enter a character to search:")
if char in text:
    print(f"'{char}' exists in the string.")
else:
    print(f"'{char}' doesnot exists in the string.")
if char not in text:
    print(f"'{char}' is not in the string.")
else:
    print(f"'{char}' is in the string.")
```

OUTPUT:

```
Enter a string:CHAHAT
Enter a character to search:A
'A' exists in the string.
'A' is in the string.
```

29.Use a combination of arithmetic and assignment operators to calculate thhe factorial of a number.

INPUT:

```
num = int(input("Enter a non-negative integer:"))
factorial = 1
for i in range(1, num+1):
    factorial *= i


print(f"The factorial of {num} is {factorial}.")
```

OUTPUT:

```
==========================================================
Enter a non-negative integer:24
The factorial of 24 is 620448401733239439360000.
```

30.Take three integers and determine which one is the largest using comparison operators.

INPUT:

```
num1 = int(input("Enter the first integer:"))
num2 = int(input("Enter the second integer:"))
num3 = int(input("Enter the third integer:"))

if num1 >= num2 and num1 >=num3:
    largest = num1
elif num2 >= num1 and num2 >= num3:
    largest = num2
else:
    largest = num3
print(f"The largest number is {largest}")
```

OUTPUT:

```
========== RESTART: C:\Users\ac
Enter the first integer:28
Enter the second integer:29
Enter the third integer:30
The largest number is 30
```

31.Create a truth table for A and B and A or B.

INPUT:

```
print("A\tB\tA and B\tA or B")
values = [True,False]
for A in values:
    for B in values:
        print(f"{A}\t{B}\t{A and B}\t{A or B}")
```

OUTPUT:

```
=========================================
A       B       A and B A or B
True    True    True    True
True    False   False   True
False   True    False   True
False   False   False   False
>>
```

32. Demonstrate short-circuit evaluation in logical operators.

INPUT:

```
def check():
    print("Function is evaluated")
    return True
print("Testing 'and' short-ciruit:")
result_and = False and check()
print(f"Result: {result_and}\n")
print("Testing 'or' short-circuit:")
result_or = True or check()
```

OUTPUT:

```
--------------------------------------------------
Testing 'and' short-ciruit:
Result: False

Testing 'or' short-circuit:
>
```

33. Evaluate: 10+3*2**2//4 and explain the order of precedence

- 10 + 3 * 2 ** 2 // 4
- ** Exponentiation (highest precedence)
- *,/,//,% Multiplication,divison, floor division , modulo (left to right)
- +,- Addition and substraction (lowest precedence)

Exponentiation:

2 ** 2 = 4

10 + 3 * 4 //4

Multiplication and floor devision(left to right):

- Multiplication: 3 * 4 = 12
- floor devision: 12 // 4 = 3

10+3

Result:

13

34.Use bitwise left and right shift operators.

INPUT:

```
File  Edit  Format  Run  Options  Window  Help
num = int(input("Enter an integer:"))
left_shift = num << 2
right_shift = num >> 2
print(f"{num} << 2 = {left_shift}")
print(f"{num} >> 2 = {right_shift}")
```

OUTPUT:

```
==========
Enter an integer:24
24 << 2 = 96
24 >> 2 = 6
>
```

35.Use ternary operator to find the minimum of two numbers.

INPUT:

```
num1 = float(input("Enter the first number:"))
num2 = float(input("Enter the second number:"))
minimum = num1 if num1 < num2 else num2
print(f"The minimum of {num1} and {num2} is {minimum}.")
```

OUTPUT:

```
--------------------------------------------------
Enter the first number:24
Enter the second number:26
The minimum of 24.0 and 26.0 is 24.0.
```

36.Create a list of 5 integers and print the list.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
print("The list of integers is:",numbers)
```

OUTPUT:

```
================================================

The list of integers is: [10, 20, 30, 40, 50]
```

37. Append an element to a list.

INPUT:

```
numbers = [10, 20, 30, 40, 50,]
numbers.append(60)
print("Updated list:", numbers)
```

OUTPUT:

```
==================================================

Updated list: [10, 20, 30, 40, 50, 60]
```

38. Insert an element at index 2 in a list.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
numbers.insert(2,25)
print("Updated list:", numbers)
```

OUTPUT:

```
==========

Updated list: [10, 20, 25, 30, 40, 50]
```

39. Remove an element from a list by value and by index.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
numbers.remove(30)
print("After removing 30 by value:", numbers)
removed_element = numbers.pop(2)
print(f"After removing element at index 2({removed_element}):",numbers)
```

OUTPUT:

```
After removing 30 by value: [10, 20, 40, 50]
After removing element at index 2(40): [10, 20, 50]
>
```

40. Sort a list of numbers in ascending and descending order.

INPUT:

```
numbers = [50, 20, 40, 10, 30]
numbers.sort()
print("Ascending order:", numbers)
numbers.sort(reverse=True)
print("Descending order:", numbers)
```

OUTPUT:

```
Ascending order: [10, 20, 30, 40, 50]
Descending order: [50, 40, 30, 20, 10]
>
```

41. Find the sum and average of elements in a list.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
total = sum(numbers)
average = total/len(numbers)
print(f"Sum of elements:{total}")
print(f"Avrage of elements:{average}")
```

OUTPUT:

```
Sum of elements:150
Avrage of elements:30.0
>
```

42.Use list slicing to get every second element.

INPUT:

```
numbers = [10, 20, 30, 40, 50, 60, 70]
every_second = numbers[::2]
print("Every second element:",every_second)
```

OUTPUT:

```
==========================================

Every second element: [10, 30, 50, 70]
>
```

43. Check if an element exists in a list.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
element = int(input("Enter a number to search:"))
if element in numbers:
    print(f"{element}exists in the list.")
else:
    print(f"{element} doesnot exists in the list.")
```

OUTPUT:

```
Enter a number to search:30
30exists in the list.
>
```

44. Reverse a list using slicing and using the reverse() method.

INPUT:

```
numbers = [10, 20, 30, 40, 50]
reverse_slice = numbers[::-1]
print("Reversed list using slicing:", reverse_slice)
numbers.reverse()
print("Reversed list using reverse():",numbers)
```

OUTPUT:

```
Reversed list using slicing: [50, 40, 30, 20, 10]
Reversed list using reverse(): [50, 40, 30, 20, 10]
```

45. Create a nested list and access elements inside the inner list.

INPUT:

```
nested_list = [1, 2, [10, 20, 30], 4, 5]
inner_list = nested_list[2]
print("Inner list:",inner_list)
print("First element of inner list:", nested_list[2][0])
print("Second element of inner list:", nested_list[2][1])
print("Third element of inner list:", nested_list[2][2])
```

OUTPUT:

```
==========================================
Inner list: [10, 20, 30]
First element of inner list: 10
Second element of inner list: 20
Third element of inner list: 30
```

46.Create a tuple of 5 items and print it.

INPUT:

```
my_tuple = (10, 20, 30, 40, 50)
print("The tuple is:",my_tuple)
```

OUTPUT:

```
========================================

The tuple is: (10, 20, 30, 40, 50)
```

47. Access the 3rd item of a tuple.

INPUT:

```
my_tuple = (10, 20, 30, 40, 50)
third_item = my_tuple[2]
print("The 3rd item of the tuple is:", third_item)
```

OUTPUT:

```
================================================

The 3rd item of the tuple is: 30

>
```

48. Convert a list into a tuple and vie versa.

INPUT:

```
my_list = [10, 20, 30, 40, 50]
my_tuple = tuple(my_list)
print("List converted to tuple:", my_tuple)
new_list = list(my_tuple)
print("Tuple converted back to list:", new_list)
```

OUTPUT:

```
List converted to tuple: (10, 20, 30, 40, 50)
Tuple converted back to list: [10, 20, 30, 40, 50]
```

49. Count the number of times a value appears in a tuple.

INPUT:

```
my_tuple = (10, 20, 30, 20, 40, 20, 50)
value = int(input("Enter a value to count:"))
count = my_tuple.count(value)
print(f"The value {value} appears {count} times in the tuple.")
```

OUTPUT:

```
Enter a value to count:20
The value 20 appears 3 times in the tuple.
```

50. Try to change an item in a tuple and observe the result.

INPUT:

```
my_tuple = (10, 20, 30, 40, 50)
try:
    my_tuple[2] = 35
except TypeError as e:
    print("Error:", e)
```

OUTPUT:

```
Error: 'tuple' object does not support item assignment
```

51. Create a set with duplicate values and print it.

INPUT:

```
my_tuple = (10, 20, 30, 40, 50)
try:
    my_tuple[2] = 35
except TypeError as e:
    print("Error:", e)
my_set = {10, 20, 30, 20, 40, 10, 50}
print("The set is:", my_set)
```

OUTPUT:

```
Error: 'tuple' object does not support item assignment
The set is: {50, 20, 40, 10, 30}
>
```

52. Add an element to a set.

INPUT:

```
my_set = {10, 20, 30, 40, 50}
my_set.add(60)
print("Updated set:", my_set)
```

OUTPUT:

```
--------------------------------------------------

Updated set: {50, 20, 40, 10, 60, 30}
```

53. Perform union, intersection and difference between two sets.

INPUT:

```
set1 = {10, 20, 30, 40}
set2 = {30, 40, 50, 60}
union_set = set1 | set2
print("union:", union_set)
intersection_set = set1 & set2
print("Intersection:", intersection_set)
difference_set = set1 - set2
print("Difference(set1 - set2):", difference_set)
```

OUTPUT:

```
================================================
union: {40, 10, 50, 20, 60, 30}
Intersection: {40, 30}
Difference(set1 - set2): {10, 20}
>
```

54. Check if a value exists in a set.

INPUT:

```
my_set = {10, 20, 30, 40, 50}
value = int(input("Enter a value to check:"))
if value in my_set:
    print(f"{value} exists in the set.")
else:
    print(f"{value} doesnot exist in the set.")
```

OUTPUT:

```
Enter a value to check:40
40 exists in the set.
```

55. Remove an item from a set using remove() and discard() and explain the difference.

INPUT:

```
my_set = {10, 20, 30, 40, 50}
my_set.remove(30)
print("After remove(30):", my_set)
my_set.discard(40)
print("After discard(40):", my_set)
my_set.discard(100)
print("After discard(100):", my_set)
```

OUTPUT:

```
After remove(30): {50, 20, 40, 10}
After discard(40): {50, 20, 10}
After discard(100): {50, 20, 10}
```

56. Create a dictionary with 3 key-value pairs.

INPUT:

```
my_dict = {
    "name" : "Chahat",
    "age"  : 18,
    "city" : "Jagraon"
}
print("The dictionary is:", my_dict)
```

OUTPUT:

```
The dictionary is: {'name': 'Chahat', 'age': 18, 'city': 'Jagraon'}
```

57. Acess and modify a value in a dictionary using its key.

INPUT:

```
my_dict = {
      "name" : "Chahat",
      "age"  : 18,
      "city" : "Jagraon"
}
print("Original age:", my_dict["age"])
my_dict["age"] = 19
print("updated dictionary:", my_dict)
```

OUTPUT:

```
Original age: 18
updated dictionary: {'name': 'Chahat', 'age': 19, 'city': 'Jagraon'}
```

58. Add a new key-value pair to an existing dictionary.

INPUT:

```
my_dict = {
      "name" : "Chahat",
      "age"  : 18,
      "city" : "Jagraon"
}
my_dict["profession"] = "student"
print("updated dictionary:", my_dict)
```

OUTPUT:

```
========== RESTART: C:\Users\acer\Documents\python21
updated dictionary: {'name': 'Chahat', 'age': 18, 'city': 'Jagraon', 'profession': 'student'}
```

59.Iterate over a dictionary and print its keys and values.

INPUT:

```
my_dict = {
    "name" : "Chahat",
    "age"  : 18,
    "city" : "Jagraon"
}
for key, value in my_dict.items():
    print(f"key:{key}, value: {value}")
```

OUTPUT:

```
========== RESTART: C:\
key:name, value: Chahat
key:age, value: 18
key:city, value: Jagraon
```

60. Use a dictionary to store and display student names and their marks.

INPUT:

```
student_marks = {
      "Chahat": 90,
      "Sourav": 79,
      "ashan" : 80,
      "ekam"  : 85
      }
for student, marks in student_marks.items():
      print(f"{student}:{marks}")
```

OUTPUT:

```
==========
Chahat:90
Sourav:79
ashan:80
ekam:85
```

THANKYOU!!!