Programming In Python
Assignment 1

Submitted To - Diksha Rani Mam

Submitted By = Prince Chand

Branch = B.Tech CSE - A

Semester = 5th

Roll No. = 1906121

Roll No. = 1906121 S.Code = BTCS 510-18

Que-1. Write a program to get a string made of the first 2 and the last 2 chars from a given a string. if the string length is less than 2, return instead the empty string.

Ans- Source Code :-

```
def string_both_ends(str):
  if len(str) < 2:
    return ''
  return str[0:2] + str[-2:]
print(string_both_ends('princechand'))
print(string_both_ends('c'))
print(string_both_ends('prince'))
```

Output :-

prnd

prce

Que-2. Write a program for type conversions of data types.

Ans- Source Code :-

```
# Python program to demonstrate
# implicit type conversion
# Python automatically converts
# a to int
```

Signature = Prince Chand

```
a = 8
print ( type (a))

# Python automatically converts b to float
b = 2.0
print ( type (b))

# Python automatically converts
# c to int as it is a floor division
c = a//b
print (c)
print (type (c))
```

Output :-

```
<class 'int'>
<class 'float'>
4.0
<class 'float'>
```

Que 1-3. Write a program to perform different operators on numbers in python.

Ans :- Source Code :-

```
# Arithmatic operators
print ('Arithmetic Operators:-')
x = 5
y = 2
print ('x+y=', x+y)    # addition operator
print ('x-y =', x-y)    # substraction operator
```

```
print('x**y=', x**y)  # multiplication operator
print('x/y =', x/y)   # division operator
print('x%y =', x%y)   # percentail operator
print('x//y =', x//y) # Floor division operator
print('x***y =', x***y) # exponent operator

# Comparison operators
print('Comparison operators:-')
a=20
b=25
print('a == b is', x==y) # equal comparison operator
print('a != b is', x!=y) # not equal comparison operator
print('a> b is', x>y)    # greater then operator
print('a< b is', x<y)    # less than operator
print('a >=b is', x>=y)  # greater than or equal to operator
print('a <=b is', x<=y)  # less than or equal to operator

# Bitwise operator
print('Bitwise operator:-')
c= 10
d = 4
print(a & b) # bitwise AND operator
print(a | b) # bitwise OR operator
print(a ^ b) # bitwise XOR operator
print(~a) # bitwise NOT operator
print(a<<2) # bitwise Right shift operator
print(a >>2) # bitwise left shift operator
```

```
# Identity operator
print ('Identity operator :-')
# Identity operator
e=6
if (type(e) is int):
    print ('true')
else:
    print ('false')
# not identity operator
f=4.5
if (type(e) is not int):
    print ('true')
else:
    print ('False')
```

Output :-

Arithmetic operators :-

x+y = 7

x-y = 3

x*y = 10

x/y = 2.5

x % y = 1

x // y = 2

x**y = 25

Comparison operators :-

a == b is False

a != b is True

a > b is True

a < b is false

a >= b is True
a <= b is false

Bitwise operators :-
16
29
13
-21
80
5

Identity operators :-
true
false

Que 4. Describe all the data type in Python with example.

Ans :- 1. Python Numbers :- Integers, floating point numbers and complex numbers fall under Python number category. eg :- int, float and complex etc.

2. Python List :- List is an ordered sequence of items. It is one of the most used datatype in Python and is very flexible. All items in a list do not need to be of the same type.
eg :- a = [1, 2.2, 'python']

3. Python Tuple :- Tuple is an ordered sequence of items same as a list. The only difference is that tuples are immutable. Tuple once created cannot be modified.
eg :- t = (5, 'program', 1+3j)

4. Python String :- string is a sequence of unicode characters. We can use single quotes or double quotes to represent strings.
eg :- S = "This is a string"

5. Python Set :- Set is an unordered collection unique items. Set is defined by values separated by comma inside braces { }.
eg :- a = {5, 2, 3, 1, 4}

6. Python Dictionary :- Dictionary is an unordered collection of key-value pairs. It is generally used when we have a huge amount of data.
eg :- d = {1: 'value', 'key': 2}

Ques 5. How to use comments in python with example.

Ans:- Comments can be used to explain Python code. Comments can be used to prevent execution when testing code. Comments starts with a # (for single line comments), """ """ (for multi line comments) and Python will ignore them.

eg:-
```
# This is a comment
print ("Hello, World!")
```

Output :- Hello, World!