PYTHON PROGRAMMING

UNIT -1 : Introduction to Python

Lec-3

Today's Target

- Operators
- Some basic program
- AKTU PYQs

Pragya Rajvanshi

B.tech,M.tech

Gateway classes

Write a program to find out the area and perimeter of Square.

```
a=float(input("enter the side of sqaure\n"))
area=a*a
perimeter=4*a
print("area of sqaure",area)
print("perimeter of sqaure",perimeter)
```

```
enter the side of sqaure
3.4
area of sqaure 11.559999999999999
perimeter of sqaure 13.6
```

GATEWAY CLASSES
Gateway Classes

Q. **How do you read an input from a user in python to be used as an integer in the rest of the program ? Explain with an example. (AKTU)**

```
a=int(input("enter the number\n"))
```

```
enter the number
3
```

Benefits of python(interpreted , high level, OOP)

- Python is interpreted, which allows for easier debugging and code development.
- Python is portable across operating systems and interactive, allowing real-time code execution and testing.
- Python is dynamically typed, meaning you don't need to declare data types explicitly.
- Python is a high-level language that abstracts low-level details, making it more user-friendly.
- Python is open source.

GATEWAY CLASSES
Gateway classes

Ques. In some language every statement ends with semi-colon (;) what happens if you put a semicolon at the end of python statement

- A semicolon in Python signifies separation rather than termination.
- No error produce.

```
1  print("hi");print("bye");print("hello")
2
```

```
hi
bye
hello
```

GATEWAY CLASSES
Gateway Classes

OPERATOR

- These are the special symbols in Python and are used to execute an Arithmetic or Logical computation
- An Operand is a value that the operator needs to complete a task.
- Types of Operators in Python-
- Arithmetic operators.
- Comparison operators .
- Assignment operators.
- Logical operators .
- Bitwise operators.

Special operators

- Identity operators
- Membership operators
- **1 Arithmetic operators**

operator	example
+,-,*	3+3=6 5-3=2 5*3=15
/,%(modulus) **(exponent)	50/2=25, 21%2=1 10**3=1000
//(floor division)	It divides two numbers and rounds down to the nearest integer. result = 7 // 3 print(result) # Output: 2

Gateway classes

2. Comparison operator

```
1  x=10
2  y=3
3  print(x+y)
4  print(x-y)
5  print(x*y)
6  print(x/y)
7  print(x%y)
8  print(x//y)
```

```
13
7
30
3.3333333333333335
1
3
```

```
print(2**3)#exponent
```

```
8
```

```
x=10
y=3
print(x<y)#less than
print(x<=y)#less than equal to
print(x>y)#greater than
print(x>=y)#greater than equal to
print(x==y)#equal to
print(x!=y)#not equal to
```

```
False
False
True
True
False
True
```

GATEWAY CLASSES
Gateway Classes

3. logical Operators

- It is used to make decision based on condition.
- And - Logical AND
- Or - Logical OR
- Not - Logical not

```
x=10>3
y=3<9
print("x and y",x and y)
print("x or y",x or y)
print("not x",not(x))
```

```
x and y True
x or y True
not x False
```

4. Bitwise operator

- Bitwise operators are used in Python to perform the operations on binary numerals or bit patterns.
- It operates bit by bit.
- & Binary AND
- | Binary OR
- ^ Binary XOR
- ~ Binary Ones Complement
- << Binary Left Shift
- >> Binary Right Shift

GATEWAY CLASSES
Gateway classes

```
print(2&5)
print(2|5)
print(2^5)
print(~5)
print(5<<1)
print(5>>1)
```

```
0
7
7
-6
10
2
```

Bitwise and

	128	64	32	16	8	4	2	1
2	0	0	0	0	0	0	1	0
5	0	0	0	0	0	1	0	1
2&5	0	0	0	0	0	0	0	0

Bitwise or

	128	64	32	16	8	4	2	1
2	0	0	0	0	0	0	1	0
5	0	0	0	0	0	1	0	1
2|5	0	0	0	0	0	1	1	1

Gateway classes

GATEWAY CLASSES

Bitwise ones' complement

	128	64	32	16	8	4	2	1
5	0	0	0	0	0	1	0	1
~5	1	1	1	1	1	0	1	0

Bitwise xor

	128	64	32	16	8	4	2	1
2	0	0	0	0	0	0	1	0
5	0	0	0	0	0	1	0	1
2^5	0	0	0	0	0	1	1	1

Gateway Classes

GATEWAY CLASSES

GATEWAY CLASSES

	128	64	32	16	8	4	2	1
5	0	0	0	0	0	1	0	1
5<<1 Bitwise left shift	0	0	0	0	1	0	1	0

	128	64	32	16	8	4	2	1
5	0	0	0	0	0	1	0	1
5>>1 Bitwise right shift	0	0	0	0	0	0	1	0

Gateway classes

Membership operator

- Membership Operators are mainly used in Python to find whether a **value or variable is present in a sequence** (string, tuple, list, set, and directory).
- **"In" and "Not In"** are the two membership operators available in Python.

main.py

```
1  l1=[1,2,3,4]
2  b=2
3  if b in l1:
4      print("hello")
5  else:
6      print("bye")
7
```

```
hello
```

GATEWAY CLASSES
Gateway classes

GATEWAY CLASSES

Membership Operator	Description	Syntax
in Operator	Returns True if the value exists in a sequence, else returns False	Value/variable in sequence
not in Operator	Returns False if the value exists in a sequence, else returns True	Value/variable not in sequence

Gateway classes

main.py

```
1  l1=[1,2,3,4]
2  b=2
3  if b  not in l1:
4      print("hello")
5  else:
6      print("bye")
```

bye

main.py

```
1  l1=(1,2,3,4)
2  b=2
3  if b  not in l1:
4      print("hello")
5  else:
6      print("bye")
```

bye

```
1  l1=(1,2,3,4)
2  b=2
3  if b in l1:
4      print("hello")
5  else:
6      print("bye")
```

hello

GW GATEWAY CLASSES
Gateway classes

```python
l1="pythonprogramming"
b="l"
if b in l1:
    print("hello")
else:
    print("bye")
```

bye

```python
s={"classA":3,"CLASSB":2,"CLASSC":7}
b="classA"
if b in s:
    print("bbbb")
else:
    print("ccccc")
```

input

bbbb

```python
l1="pythonprogramming"
b="l"
if b not in l1:
    print("hello")
else:
    print("bye")
```

hello

```python
s={"classA":3,"CLASSB":2,"CLASSC":7}
b="classA"
if b not in s:
    print("bbbb")
else:
    print("ccccc")
```

input

ccccc

Gateway classes

Identity Operators

Identity operators are used to compare the objects if both the objects are actually of the same data type and share the same memory location.

Identity Operator	Description	Syntax
is Operator	Returns True if both objects refers to same memory location, else returns False	obj1 is obj2
is not Operator	Returns False if both object refers to same memory location, else returns True	obj1 is not obj2

GATEWAY CLASSES
Gateway classes

Variable

```
x=2
y=5
print(type(x))
print(type(y))
print(id(x))
print(id(y))
print(x is y)
print(x is not y)
```

```
<class 'int'>
<class 'int'>
1271795638602 40
1271795638603 36
False
True
```

Variable

```
x=2
y=2
print(type(x))
print(type(y))
print(id(x))
print(id(y))
print(x is y)
print(x is not y)
```

```
<class 'int'>
<class 'int'>
133606456836368
133606456836368
True
False
```

GATEWAY CLASSES
Gateway Classes

Tuple

```
x=(1,2,3,4)
y=(1,2,3,4)
print(type(x))
print(type(y))
print(id(x))
print(id(y))
print(x is y)
print(x is not y)
```

```
<class 'tuple'>
<class 'tuple'>
134520851275968
134520851275968
True
False
```

string

```
x="hello"
y="hello"
print(type(x))
print(type(y))
print(id(x))
print(id(y))
print(x is y)
print(x is not y)
```

```
<class 'str'>
<class 'str'>
129451994184752
129451994184752
True
False
```

GATEWAY CLASSES
Gateway classes

List

```
x=[1,2,3,4]
y=[1,2,3,4]
print(type(x))
print(type(y))
print(id(x))
print(id(y))
print(x is y)
print(x is not y)
```

```
<class 'list'>
<class 'list'>
134489793223744
134489793300672
False
True
```

GATEWAY CLASSES
Gateway classes



Gateway Classes