Operators in Python

Operators are the symbol that is used to perform operations on values variables.

name = “R.Rodrick”

Here equal to (=) sign is an operator.

Common arithmetic operators:

Addition	+
Subtraction	-
Multiplication	*
Exponent	**
Division	/
Floor Division	//
Remainder	%

Examples:

```
x =5
result = x+10
print(result)
```

output: 15

```
x = 5
quotient = x//2
remainder = x%2
print(“Quotient is:”, quotient)
print(“Reminder is:”, remainder)
```

Output: Quotient is: 2

Remainder is: 1

```
number = 34*5-9/3
number = (34*5)-(9/3)
print(number)
```

(+) operator can be used for concatenation.

```
str1 = “Hi”
str2 = “John”
print(str1+str2)
```

Assignment operators:

Assignment operators are used in Python to assign values to variables.

`a = 5` is a simple assignment operator that assigns the value 5 on the right to the variable `a` on the left.
There are various compound operators in Python like `a += 5` that adds to the variable and later assigns the same. It is equivalent to `a = a + 5`.

Examples:

```
x,y = 5,6
```

equivalent to

```
x = 5
y = 6
```

More assignment operators’ examples:

```
x+=10     # x = x+5
```

```
x- = 10    # x = x-5
```

Program-1: Suppose you are a school student, and you need to pay taka 5300 tuition fee for the next month. The school is giving you a discount of 10% on the early payment of your tuition fee. Since it's a good offer, you decided to move on early payment. Can you find out how much money you have to pay?

Solution:

```
fee = 5300    or  fee = float(input("Enter your fee: "))
discount_percent = 10
discount_amount = (discount_percent)/100*fee
discounted_fee = fee -discount_amount
print ("Fee after discount:", discounted_fee)
```

Program-2: Can you create a program to convert distant in kilometers to miles?

Hints: 1 mile = 0.621371 km

Solution:

```
kilometers = float(input("Enter kilometers to convert: "))
miles = kilometers * 0.621371
print ("The conversion of Kilometres to miles is: ", miles)
```

Important points to remember:

1) Equals operators (=) assigns the value in the right to the variable in the left.
2) Arithmetic operators perform basic arithmetic operations: such as addition, subtraction, division, multiplication etc.
3) If we use the + operator with string, it concatenates two strings.
4) To make out code more readable , we can use the parenthesis, for example: 34*(15-5)

Python Booleans:

In programming you often need to know if an expression is True or False.

You can evaluate any expression in Python, and get one of two answers, True or False.

When you compare two values, the expression is evaluated and Python returns the Boolean answer:

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```

x = "Hello"

y = 15

print(bool(x))

print(bool(y))

result1 = True

result2 = False

print(result1)

print(result2)

Now using comparison operators:

number = 5

print(number<10)

output: True

Python Comparison Operators:

Comparison operators are used to compare two values:

Operator	Name	Example

==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Examples:

Ex-1:

x = 5

y = 3

print(x == y)

Output: False

Ex-2:

x = 5

y = 3

print(x != y)

Output: True

Ex-3:

x = 5

```
y = 3

print(x > y)
```

Output: True

returns True because 5 is greater than 3

Ex-4:

```
x = 5

y = 3

print(x < y)
```

Output: False

returns False because 5 is not less than 3

Ex-5:

```
x = 5

y = 3

print(x >= y)
```

Output: True

returns True because five is greater, or equal, to 3

Python Logical Operators:

Logical operators are used to combine conditional statements:

and	True if both operands/statements are true
or	True if either of the operands/statements are true
not	True if the operands is false.

Examples:

Ex-1:

```
age = 22
gpa = 3.8
result = age>=18 and gpa>3.6
print(result)
```

output: True

Here both conditions are true.

Ex-2:

```
age = 16
gpa = 3.8
result = age>=18 or gpa<3.6
print(result)
```

output: False

Ex-3:

```
x = 5
print(x > 3 and x < 10)
```

Output: Ture

returns True because 5 is greater than 3 AND 5 is less than 10

Ex-4:

```
x = 5
print(x > 3 or x < 4)
```

Output: True

returns True because one of the conditions are true (5 is greater than 3, but 5 is not less than 4)

Ex-5:

```
x = 5
```

print(not(x > 3 and x < 10))

Output: False

returns False because not is used to reverse the result

Python challenge:

What is the output of the following program:

```
language = “Python”
print (“1.”, language == “Python”)
age= 18
print(“2.”, age>=18)
print(“3.”, age>18)
print(“4.”, age>=18 and language == “Java”)
```

Find the output of the program using python IDE

and write the output here.

Python Identity Operators:

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location.

Examples:

```
x = ["apple", "banana"]
y = ["apple", "banana"]
z = x
print(x is z)
# returns True because z is the same object as x
print(x is y)
# returns False because x is not the same object as y, even if they have the same content
print(x == y)
```

to demonstrate the difference betweeen "is" and "==": this comparison returns True because x is equal to y

Example: 2

```
x = ["apple", "banana"]
y = ["apple", "banana"]
z = x
print(x is not z)
# returns False because z is the same object as x
print(x is not y)
# returns True because x is not the same object as y, even if they have the same content
print(x != y)

# to demonstrate the difference betweeen "is not" and "!=": this comparison returns False because x is equal to y
```

Python Membership Operators:

Membership operators are used to test if a sequence is presented in an object.

Example: 1

```
x = ["apple", "banana"]
print("banana" in x)
# returns True because a sequence with the value "banana" is in the list
```

Example: 2

```
x = ["apple", "banana"]
print("pineapple" not in x)
# returns True because a sequence with the value "pineapple" is not in the list
```

Important Points to remember:

1) The Boolean represents one of two values: either true or false.

2) The comparison operators are used to compare two values.
3) If the comparison is right, the result is true. If not, the result is false.
4) The logical operators are used on Booleans.
5) There are three logical operators: and, or and not.