Valluru Sai Balaji

AF0401675

LAB-I:PYTHON OPERATORS

Q.1 Write a python program for arithmetic operators

# Python program for Arithmetic Operators

Num1= 15

Num2= 4

# Addition

Addition = Num1+ Num2

print(Addition)

output

19

# Subtraction

Subtraction = Num1-Num2

print(Subtraction)

output

11

# Multiplication

Multiplication = Num1\*Num2

print(Multiplication)

output

60

# Division

Division = Num1/Num2

print(Division)

output

3.75

# Modulus

Modulus = Num1 % Num2

print(Modulus)

output

3

# Exponentiation

Exponentiation = Num1\*\*Num2

print(Exponentiation)

output

50625

# Floor Division

Floor\_division = Num1// Num2

print(Floor\_division)

output

3

Explanation

1. **Addition**: 15 + 4 = 19
2. **Subtraction**: 15 - 4 = 11
3. **Multiplication**: 15 \* 4 = 60
4. **Division**: 15 / 4 = 3.75 (Floating-point division)
5. **Modulus**: 15 % 4 = 3 (Remainder when 15 is divided by 4)
6. **Exponentiation**: 15 \*\* 4 = 50625 (15 raised to the power of 4)
7. **Floor Division**: 15 // 4 = 3 (Division that returns the largest integer less than or equal to the result)

Q.2 Write a python program for assignment operators

# Python program for Assignment Operators

Num1 = 10

Num2 = 3

# Add and assign

Num1 += Num2 # Equivalent to Num1 = Num1 + Num2

print("Num1 += Num2 is", Num1)

output

Num1 += Num2 is 13

# Subtract and assign

Num1 -= Num2 # Equivalent to Num1 = Num1 - Num2

print("Num1 -= Num2 is", Num1)

output

Num1 -= Num2 is 10

# Multiply and assign

Num1 \*= Num2 # Equivalent to Num1 = Num1 \* Num2

print("Num1 \*= Num2 is", Num1)

output

Num1 \*= Num2 is 30

# Divide and assign

Num1 /= Num2 # Equivalent to Num1 = Num1 / Num2

print("Num1 /= Num2 is", Num1)

output

Num1 /= Num2 is 10.0

# Modulus and assign

Num1 %= Num2 # Equivalent to Num1 = Num1 % Num2

print("Num1 %= Num2 is", Num1)

output

Num1 %= Num2 is 1.0

# Exponentiation and assign

Num1 \*\*= Num2 # Equivalent to Num1 = Num1 \*\* Num2

print("Num1 \*\*= Num2 is", Num1)

output

Num1 \*\*= Num2 is 1.0

# Floor division and assign

Num1 //= Num2 # Equivalent to Num1 = Num1 // Num2

print("Num1 //= Num2 is", Num1)

output

Num1 //= Num2 is 0.0

**Explanation:**

1. **Add and assign**: Num1 += Num2 adds Num2 to Num1 and assigns the result to Num1. After this operation, Num1 becomes 13.
2. **Subtract and assign**: Num1 -= Num2 subtracts Num2 from Num1, resulting in Num1 becoming 10.
3. **Multiply and assign**: Num1 \*= Num2 multiplies Num1 by Num2, resulting in Num1 becoming 30.
4. **Divide and assign**: Num1 /= Num2 divides Num1 by Num2, resulting in Num1 becoming 10.0 (a floating-point number).
5. **Modulus and assign**: Num1 %= Num2 assigns the remainder of Num1 divided by Num2 to Num1, resulting in 1.0.
6. **Exponentiation and assign**: Num1 \*\*= Num2 raises Num1 to the power of Num2, resulting in 1.0.
7. **Floor division and assign**: Num1 //= Num2 performs floor division on Num1 by Num2, resulting in 0.0.

Q.3Write a python  program for Bitwise operators

# Python program for Bitwise Operators

Num1 = 10 # Num1 in binary: 1010

Num2 = 4 # Num2 in binary: 0100

# Bitwise AND

# Performs a bitwise AND operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 only if both corresponding bits are 1.

bitwise\_and = Num1 & Num2

print(f"Bitwise AND: {Num1} & {Num2} = {bitwise\_and}")

# Bitwise OR

# Performs a bitwise OR operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 if at least one of the corresponding bits is 1.

bitwise\_or = Num1 | Num2

print(f"Bitwise OR: {Num1} | {Num2} = {bitwise\_or}")

# Bitwise XOR

# Performs a bitwise XOR (exclusive OR) operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 only if the corresponding bits are different.

bitwise\_xor = Num1 ^ Num2

print(f"Bitwise XOR: {Num1} ^ {Num2} = {bitwise\_xor}")

# Bitwise NOT

# Performs a bitwise NOT operation on Num1.

# It inverts all the bits of Num1, changing 1s to 0s and 0s to 1s.

bitwise\_not = ~Num1

print(f"Bitwise NOT: ~{Num1} = {bitwise\_not}")

# Bitwise right shift

# Performs a right shift operation on Num1.

# It shifts the bits of Num1 to the right by 1 position, effectively dividing the number by 2.

right\_shift = Num1 >> 1

print(f"Right Shift: {Num1} >> 1 = {right\_shift}")

# Bitwise left shift

# Performs a left shift operation on Num1.

# It shifts the bits of Num1 to the left by 1 position, effectively multiplying the number by 2.

left\_shift = Num1 << 1

print(f"Left Shift: {Num1} << 1 = {left\_shift}")

output

Bitwise AND: 10 & 4 = 0

Bitwise OR: 10 | 4 = 14

Bitwise XOR: 10 ^ 4 = 14

Bitwise NOT: ~10 = -11

Right Shift: 10 >> 1 = 5

Left Shift: 10 << 1 = 20

**Explanation:**

1. **Bitwise AND** (10 & 4):
   * Binary of 10: 1010
   * Binary of 4: 0100
   * Result (AND): 0000 (Decimal: 0)
2. **Bitwise OR** (10 | 4):
   * Binary of 10: 1010
   * Binary of 4: 0100
   * Result (OR): 1110 (Decimal: 14)
3. **Bitwise XOR** (10 ^ 4):
   * Binary of 10: 1010
   * Binary of 4: 0100
   * Result (XOR): 1110 (Decimal: 14)
4. **Bitwise NOT** (~10):
   * Binary of 10: 1010
   * Result (NOT): Inverting all bits results in ...11110101 (which is -11 in two's complement form)
5. **Right Shift** (10 >> 1):
   * Binary of 10: 1010
   * Right Shift by 1 bit: 0101 (Decimal: 5)
6. **Left Shift** (10 << 1):
   * Binary of 10: 1010
   * Left Shift by 1 bit: 10100 (Decimal: 20)

Q.4 Write a python program to calculate greatest of three numbers

# python program to calculate greatest of three numbers

# python program to calculate greatest of three numbers

num1 = 10

num2 = 15

num3 = 20

if num1 > num2 and num1 > num3:

greatest = num1

elif num2 > num1 and num2 > num3:

greatest = num2

else:

greatest = num3

print(greatest)

output

20

**Explanation:**

* The program compares num1, num2, and num3.
* If num1 is greater than both num2 and num3, it is assigned as the greatest.
* If num2 is greater than both num1 and num3, it becomes the greatest.
* Otherwise, num3 is the greatest.