Answers

1).

```
# Define two integer variables
num1 = 20
num2 = 5

# Perform arithmetic operations and store results
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2

# Print the formatted output
print(f"First variable is {num1} & second variable is {num2}.")
print(f"Addition: {num1} + {num2} = {addition}")
print(f"Subtraction: {num1} - {num2} = {subtraction}")
print(f"Multiplication: {num1} * {num2} = {multiplication}")
print(f"Division: {num1} / {num2} = {division}")
```

2. answer

i) `/` vs `//`:

- `/` is the division operator in Python and returns a float value, even if the result is a whole number.
- `//` is the floor division operator, also known as integer division, which returns the integer part of the quotient without considering the decimal part.

(ii) `**` vs `^`:

- `**` is the exponentiation operator in Python, used to raise a number to the power of another.
- `^` is the bitwise XOR operator, used for bitwise exclusive OR operation.

3. Answer

The logical operators in Python are:

`and`: Returns True if both operands are True, otherwise returns False.
`or`: Returns True if at least one of the operands is True, otherwise returns False.
`not`: Returns True if the operand is False, and False if the operand is True.

4. Answer

Right Shift Operator (`>>`):

- The right shift operator shifts the bits of a number to the right by a specified number of positions.
- It fills the vacated bits on the left with the sign bit (the leftmost bit, which indicates the sign of the number).
- The general syntax is `x >> y`, where `x` is the number to be shifted, and `y` is the number of positions to shift.
- Example:

```
x = 10  # Binary: 1010
y = 2
result = x >> y  # Result: 2 (Binary: 0010)
```

Left Shift Operator (`<<`):

- The left shift operator shifts the bits of a number to the left by a specified number of positions.
- It fills the vacated bits on the right with zeros.
- The general syntax is `x << y`, where `x` is the number to be shifted, and `y` is the number of positions to shift.
- Example:

```
x = 5  # Binary: 0101
y = 2
result = x << y  # Result: 20 (Binary: 10100)
```

5. Answer

```
# Create a list containing integer data
my_list = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]

# Check if 10 is present in the list
if 10 in my_list:
    print("10 is present in the list.")
else:
    print("10 is not present in the list.")
```