**Q.1. Create two int type variables, apply addition, subtraction, division and multiplications and store the results in variables. Then print the data in the following format by calling the variables:**

**First variable is \_\_ & second variable is \_\_.**

**Addition: \_\_ + \_\_ = \_\_**

**Subtraction: \_\_ - \_\_ = \_\_**

**Multiplication: \_\_ \* \_\_ = \_\_**

**Division: \_\_ / \_\_ = \_\_**

Ans.) A screenshot of a computer program

Description automatically generated

**Q.2. What is the difference between the following operators:**

**(i) ‘/’ & ‘//’**

**(ii) ‘\*\*’ & ‘^’**

Ans.) **(i) ‘/’ & ‘//’: / (Forward Slash)**: The / operator is used for division in Python. It performs normal division and returns a floating-point result, regardless of the input types.

Ex.) A white rectangular object with a black border

Description automatically generated

**// (Double Slash)**: The // operator is used for floor division or integer division in Python. It performs division and rounds down the result to the nearest integer. The result is always an integer, regardless of the input types.

Ex.) A white rectangular object with a black border

Description automatically generated

**(ii) ‘\*\*’ & ‘^’: \*\* (Double Asterisk)**: The \*\* operator is used for exponentiation in Python. It raises the left operand to the power of the right operand.

Ex.) A white rectangular object with black lines

Description automatically generated

**^ (Caret or XOR Operator)**: The ^ operator is not used for exponentiation in Python. Instead, it is the bitwise XOR operator, which performs a bitwise exclusive OR operation on two integers. It compares the binary representation of each bit and returns 1 if the bits are different, and 0 if they are the same.

Ex.) A white background with a black border

Description automatically generated

**Q.3. List the logical operators.**

Ans.) There are 3 types of operators as follows:

1. **AND**: AND operator returns True when both operands TRUE. If any one of the operands doesn’t holds condition true, then Output is FALSE.

|  |  |  |
| --- | --- | --- |
| Statement 1 | Statement 2 | Result |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

|  |  |  |
| --- | --- | --- |
| Statement 1 | Statement 2 | Result |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

1. **OR**: OR operator returns TRUE when any one of the operands holds condition True. If both operands don’t hold condition true, then Output is FALSE.

|  |  |  |
| --- | --- | --- |
| Statement 1 | Statement 2 | Result |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

|  |  |  |
| --- | --- | --- |
| Statement 1 | Statement 2 | Result |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

1. **NOT**: NOT is inversion. It takes single input and returns single output with inversion. i.e. if Input is TRUE then Output is FALSE.

|  |  |
| --- | --- |
| 1. Statement | Result |
| TRUE | FALSE |
| FALSE | TRUE |

|  |  |
| --- | --- |
| Statement | Result |
| 1 | 0 |
| 0 | 1 |

**Q.4. Explain right shift operator and left shift operator with examples.**

Ans.) **Right Shift Operator (>>):** The right shift operator shifts the bits of an integer to the right by a specified number of positions. The rightmost bits are discarded, and the leftmost bits are filled with the original value of the leftmost bit (0 for non-negative numbers and 1 for negative numbers).

**Syntax: result = number >> n**

Here, ‘**number**’ is the integer whose bits are to be shifted, and “**n’** is the number of positions to shift the bits.

Ex.)

A screenshot of a computer

Description automatically generated

For 4 Binay number is (00100)

2.) **Left Shift Operator (<<):**The left shift operator shifts the bits of an integer to the left by a specified number of positions. The leftmost bits are discarded, and the right most bits are filled with zeros.

**Syntax:** **result = number << n**

Here, ‘**number’** is the integer whose bits are to be shifted, and **‘n’** is the number of positions to shift the bits.

Ex.)

**A screenshot of a computer code

Description automatically generated**

**Q.5. Create a list containing int type data of length 15. Then write a code to check if 10 is present in the list or not.**

Ans.) A screenshot of a computer code

Description automatically generated

**A screenshot of a computer code

Description automatically generated**