**21 May**

**Python Basic – 2**

**Assignment - 7**

**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) ‘/’ & ‘//’**

**Ans:** ‘/’ operator is known as floating point division while ‘//’ operator is known as floor division.

* ‘/’ operator: It returns the exact quotient of the division, including decimal places. The result is always a float data type. E.g. ‘5 / 2’ results ‘2.5’
* ‘//’ operator: It returns the quotient of division which is rounded number. It discards the decimal part of the output. E.g. ‘5 // 2’ results ‘2’ as the ‘0.5’ decimal point is eliminated.

A white rectangular sign with black text

Description automatically generated

**(ii) ‘\*\*’ & ‘^’**

**Ans: \*\* Operator (Exponentiation):**

* The \*\* operator is used for exponentiation, which raises a number to a power.
* It calculates the result of raising the base number to the exponent.
* For example, 2 \*\* 3 calculates 2 raised to the power of 3, resulting in 8.

**^ Operator (Bitwise XOR):**

* The ^ operator is used for bitwise XOR (exclusive OR) operation.
* It performs the XOR operation bit by bit between the binary representations of two numbers.
* For example, 5 ^ 3 performs the XOR operation between the binary representations of 5 and 3, resulting in 6.

Example:

A white rectangular object with black legs

Description automatically generated

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

**Ans:** In Python, the logical operators are used to perform logical operations on Boolean values (True or False) or expressions. The logical operators in Python are:

1. **and Operator:**

* The and operator returns True if both operands are True, otherwise it returns False.
* It evaluates the second operand only if the first operand is True.
* Example: (True and False) = False.

1. **or Operator:**

* The or operator returns True if at least one of the operands is True, otherwise it returns False.
* It evaluates the second operand only if the first operand is False.
* Example: (True or False) = True.

1. **not Operator:**

* The not operator returns the opposite of the operand's logical value.
* It negates the boolean value of the operand, i.e., if the operand is True, not returns False, and if the operand is False, not returns True.
* Example: (not True) = False.

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

**Ans**: In Python, the right shift (>>) and left shift (<<) operators are bitwise shift operators. They manipulate the bits of integers, shifting them to the right or left, respectively.

1. **Right Shift Operator (>>):**

* The right shift operator shifts the bits of the first operand to the right by the number of positions specified by the second operand.
* Each shift to the right effectively divides the value by 2, discarding the rightmost bit and shifting in zeros from the left.
* Example:

A white rectangular object with a black border

Description automatically generated

1. **Left Shift Operator (<<):**

* The left shift operator shifts the bits of the first operand to the left by the number of positions specified by the second operand.
* Each shift to the left effectively multiplies the value by 2, shifting in zeros from the right.
* Example:

A white rectangular object with black text

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**