**Assignment\_2:**

1. What are the two values of the Boolean data type? How do you write them?

🡪 1 and 0 are two values of Boolean data type. 1 represents True and 0 represents False.

Boolean values can be defined with the keywords True and False

2. What are the three different types of Boolean operators?

🡪 Three different Boolean operators:

* and: This operator require two operands and returns True if both are True
* or: This operator require two operands and returns True if one of them is True
* not: This operator require one operand and returns True only if the operand is False. It is a negation operator.

3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluates).

🡪 Truth table of **AND** operator

|  |  |  |
| --- | --- | --- |
| Operator 1 | Operator 2 | Output |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

🡪 Truth table of **OR** operator

|  |  |  |
| --- | --- | --- |
| Operator 1 | Operator 2 | Output |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

🡪 Truth table of **NOT** operator

|  |  |
| --- | --- |
| Operator 1 | Output |
| True | False |
| False | True |

4. What are the values of the following expressions?

(5 > 4) and (3 == 5)

* **False**

not (5 > 4)

* **False**

(5 > 4) or (3 == 5)

* **True**

not ((5 > 4) or (3 == 5))

* **False**

(True and True) and (True == False)

* **False**

(not False) or (not True)

* **True**

5. What are the six comparison operators?

🡪 **Comparison operators:**

* >
* <
* ==
* !=
* >=
* <=

6. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one.

🡪 equal to (‘==’) is a comparison operator and returns True if both the values are equal otherwise returns False.

Assignment operator (‘=’): is used to assign the value to the variable and it will not return anything.

Consider a situation where you want to separate the numbers from the list as add or even and store them separately.

First step we perform is takeout the number from the list and divide by two and compare the remainder is zero or not and if it is zero, it will stored in even list and if it is not it will be stored in odd list.

So, here we will be using ‘==’ operator to compare the remainder is zero when the number is divided by zero and we will be ‘=’ operator to store the value in even list or odd list.

7. Identify the three blocks in this code:

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

🡪

**First block:**

if spam == 10:

print('eggs')

**Second block:**

if spam > 5:

print('bacon')

**Third block:**

else:

print('ham')

print('spam')

print('spam')

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

**Program:**

spam=int(input('Enter the value of spam: '))

if spam == 1:

print('Hello')

elif spam == 2:

print('Howdy')

else:

print('Greetings')

9. If your programme is stuck in an endless loop, what keys you’ll press?

🡪 **CTRL+C** key is used to stop the execution of endless loop.

10. How can you tell the difference between break and continue?

🡪 break and continue statements alter the flow of execution of loop.

When break statement is executed inside the loop, it terminates the execution of loop and control of the program flows to the immediate statement after the body of the loop.

When continue statement is executed inside the loop, it skip the execution of rest of the code below the continue statement for the current iteration only and control of the program flows to the next iteration.

11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

🡪 range function returns the sequence of integers between the given range.

range(10), range(0,10), range(0,10,1): All these three functions in a for loop iterate 10 times from 0 to 9.

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

🡪for loop:

for i in range(1,11):

print(i)

🡪while loop:

x=1

while(x<11):

print(x)

x+=1

13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?

🡪 spam.bacon()