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

Ans: Boolean data type has only two values:**True and False**. (Boolean is capitalized because the data type is named after mathematician George Boole.)

- They are written as

- boolA =True

- boolB = False

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

Ans: AND, OR, and NOT

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 evaluate ).

Ans: andList **=** [[0, 0 ,0],[0,1,0],[1,1,1]]

print("\nThe AND operator truth table")

print(" A B AandB")

**for** i **in** andList:

print(i)

orList **=** [[0, 0 ,0],[0,1,1],[1,1,1]]

print("\nThe OR operator truth table")

print(" A B AorB")

**for** i **in** orList:

print(i)

notList **=** [[0, 1],[1,0]]

print("\nThe NOT operator truth table")

print(" A notA")

**for** i **in** notList:

print(i)

when you run it:

The AND operator truth table

A B AandB

[0, 0, 0]

[0, 1, 0]

[1, 1, 1]

The OR operator truth table

A B AorB

[0, 0, 0]

[0, 1, 1]

[1, 1, 1]

The NOT operator truth table

A notA

[0, 1]

[1, 0]

4. What are the values of the following expressions?

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

Ans: False

not (5 > 4)

Ans: False

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

Ans: True

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

Ans: False

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

Ans: False

(not False) or (not True)

Ans: True

5. What are the six comparison operators?

Ans: **equal to, not equal to, greater than, greater than or equal to, less than, and less than or equal to.**

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

Ans: Assignment operator "="

It is an assignment operator.

It is used for assigning the value to a variable.

example a = 10, str = "Bhanu"

Constant term cannot be placed on left hand side.

Example: 1=x; is invalid.

Equal to Operator "=="

It is a relational or comparison operator

It is used for comparing two values. It returns 1 if both the values are equal otherwise returns 0.

Constant term can be placed in the left hand side.

Example: 1==1 is valid and returns 1.

example :

x=10

y=10

if x==y:

print("True")

else:

print("False")

x**=**10

y**=**10

**if** x**==**y:

print("True")

**else**:

print("False")

True

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')

Ans: spam **=** 0 *# ist block*

**if** spam **==** 10: *# second block*

print("eggs")

**if** spam **>** 5: *# 3rd Block*

print("bacon")

**else**:

print("ham")

print("spam")

print("spam")

when you run it:

ham

spam

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.

Ans:

spam = int(input("Enter spam value :"))

if spam == 1 :

print("Hi")

elif spam == 2:

print("Hello")

else:

print("Good morning!!!")

Enter spam value :5

Good morning!!!

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

Ans: Press CTRL + C to exit from a infinite loop

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

Ans: Break Statement :

Break statements exist in Python to exit or “break” a for or while conditional loop. When the loop ends, the code picks up

from and executes the next line immediately following the loop that was broken

Continue Statement:

The continue statement is used to skip code within a loop for certain iterations of the loop. After the code is skipped,

the loop continues where it left off

*#Break Example*

lst **=** [1,2,3]

cnt **=** 0

**for** i **in** lst:

cnt**+=**lst[i]

print(cnt)

**if** cnt **==** 2:

**break**

*#In this example, the loop will break after the count is equal to 2.*

When you run it:

2

*# continue example*

**for** x **in** range(4):

**if** (x**==**2):

**continue**

print(x)

*#This example would print all numbers from 0-4 except 2*

When you run it:

0

1

3

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

Ans: There is no difference in the range(10), range(0,10) and range(0,10,1). range by default takes 0 as starting point and 1 as by default increment.

**for** i **in** range(10):

print(i)

0

1

2

3

4

5

6

7

8

9

**for** i **in** range(0,10):

print(i)

0

1

2

3

4

5

6

7

8

9

**for** i **in** range(0,10,1):

print(i)

0

1

2

3

4

5

6

7

8

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.

Ans: *# Using for loop*

**for** i **in** range(1,11):

print(i)

1

2

3

4

5

6

7

8

9

10

*# Using while loop*

cnt **=** 1

**while** cnt **<=**10:

print(cnt)

cnt**+=**1

1

2

3

4

5

6

7

8

9

10

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

Ans: import spam

a = spam.bacon()