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

The two values of the Boolean data type are typically referred to as "true" and "false".

They represent the two possible states of logic: true signifies a condition is met or a statement is valid, while false signifies a condition is not met or a statement is invalid.

In python we write it as True, False, and they are reserved keywords and are case-sensitive.

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

AND Operator (&& or AND): This operator returns true if both operands are true. Otherwise, it returns false. In most programming languages, the && symbol is used for the AND operator.

OR Operator (|| or OR): This operator returns true if at least one of the operands is true. It returns false only if both operands are false. In most programming languages, the || symbol is used for the OR operator.

NOT Operator (! or NOT): This operator negates the value of its operand. If the operand is true, it returns false. If the operand is false, it returns true. In most programming languages, the ! symbol is used for the NOT 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 evaluate).

- AND Operator

A	B	A AND B
True	True	True
True	False	False
False	True	False
False	False	False

- NOT Operator

A	B
True	False
False	True

- OR Operator

A	B	A AND B
True	True	True
True	False	True
False	True	True
False	False	False

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

4. What are the six comparison operators?

i. Equal to (==): Checks if the values of two operands are equal.

ii. Not equal to (!=): Checks if the values of two operands are not equal.

iii. Greater than (>): Checks if the value of the left operand is greater than the value of the right operand.

iv. Less than (<): Checks if the value of the left operand is less than the value of the right operand.

v. Greater than or equal to (>=): Checks if the value of the left operand is greater than or equal to the value of the right operand.

vi. Less than or equal to (<=): Checks if the value of the left operand is less than or equal to the value of the right operand.

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

The equal to operator (==) is used for comparison to check whether the values on both sides of the operator are equal. Whereas, the assignment operator (=) is used to assign a value to a variable.

Ex: a = 7 assigns the value 7 to the variable a

Whereas a == 5 checks whether the value of already initialized variable a is 5. It will return True if the value of a is actually 5, otherwise False. So in this case the output would be False, with variable a initialized as a = 7.

7. Identify the three blocks in this code:

```
spam = 0                    #Block 1: Main Block
if spam == 10:              #Block 2: Conditional Block
print('eggs')
if spam > 5:                #Block 3: Conditional Block
print('bacon')
else:                       #Block 3: Conditional Block (continuation)
print('ham')
#The below are not part of conditional block, but part of main block
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.

```
spam =  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

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

break and continue are both control flow statements used in loops to alter the flow of execution. However, they serve different purposes:

The break statement is used to immediately exit a loop. When a break statement is encountered inside a loop, the loop is terminated, and the program continues executing from the next statement after the loop. It is commonly used to prematurely exit a loop based on certain conditions.

The continue statement is used to skip the current iteration of a loop and proceed to the next iteration. When a continue statement is encountered inside a loop, the current iteration of the loop is terminated, and the loop proceeds to the next iteration. It is commonly used to skip certain iterations of a loop based on certain conditions.

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

In Python, the range() function is used to generate a sequence of numbers. range(10) is a shorthand notation that starts from 0 and generates numbers up to, but not including, the specified end point. range(0, 10) explicitly specifies both the starting and ending points of the sequence. range(0, 10, 1) explicitly specifies the starting point, ending point, and step size of the sequence.

But since 0 is the default starting point, and 1 the default step size, all the range functions: range(10), range(0, 10), and range(0, 10, 1) will produce the same output of a sequence of numbers from 0-9 I.e.: 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.

```
# Using a for loop

for i in range(1, 11):

   print(i)

# Using a while loop

i = 1

while i <= 10:

   print(i)

   i += 1
```

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

```
Import spam

Spam.bacon()
```