1.

Solution:

The two values of Boolean type are: True and False. They are written as: True and False.

Q2.

Solution: Three types of Boolean operators are: *and*, *or* and *not.*

Q3.

Solution:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **and** | | | **or** | | | |
| **Input1** | **Input2** | **output** | **Input1** | **Input2** | **output** | |
| False | False | False | False | False | False |
| False | True | False | False | True | True |
| True | False | False | True | False | True |
| True | True | True | True | True | True |

|  |  |
| --- | --- |
| **not** | |
| **input** | **output** |
| False | True |
| True | False |

Q4.

Solution:

|  |  |
| --- | --- |
| **Boolean Expression** | **Output** |
| (5 > 4) and (3 == 5) | False |
| not(5>4) | False |
| (5 > 4) or (3 == 5) | True |
| not((5 > 4) and (3 == 5)) | True |
| (True and True) and (True == False) | False |
| (not False) or (not True) | True |

Q5.

Solution: The six comparison operators are: >, <, ==, >=, <=, and != .

Q6.

Solution: The '=' is the so-called assignment operator and is used to assign the result of the expression on the right side of the operator to the variable on the left side. The '==' is the so-called equality comparison operator and is used to check whether the two expressions on both sides are equal or not. The ‘=’ or the assignment operator is used to assign/re-assign a value to a variable. For example, to assign an integer value of 10 to variable x, x = 10. The ‘==’ (comparison operator) is used to compare two entities (variables, constants, expressions or even some functions) to check whether they are equal or not. For example, to check if variable x is equal to value 10, x == 10.

Q7.

Solution:

spam = 0

if spam == 10: #Block 1

print(‘eggs’)

if spam > 5: #Block 2

print(‘bacon’)

else: #Block 3

print(‘spam’)

print(‘spam’)

print(‘spam’)

Q8.

Solution:

spam = int(input())

if spam == 1:

print(‘Hello’)

elif spam == 2:

print(‘Howdy’)

else:

print(‘Greetings!’)

Q9.

Solution: Press Ctrl + C to stop the endless loop.

Q10.

Solution: The break statement results in the immediate exit from the enclosing loop. On the other hand, the continue statement begins the next iteration of the enclosing loop.

Q11.

Solution: In a for loop, there is no difference in the execution of the for loop by writing range(10), range(0,10) or range(0,10,1). The three values in the range(0,10,1) depict the starting index 0, the end index 10 (exclusive) and the step 1. The default value for starting index and step is 0 and 1, respectively. So, writing range(10), range(0,10) or range(0,10,1) means the same thing.

Q12.

Solution:

#Print 1 to 10 using for loop

for I in range(1,11):

print(i)

#print 1 to 10 using while loop

i=1

while(i<11):

print(i)

i += 1

Q13.

spam.bacon()