**ASSIGNMENT – 2**

**TARUNCHAUDHARY358@GMAIL.COM**

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

Ans.) The two values of Boolean datatype are ‘True’ and ‘False’ and these values are assigned to a variable. They are case-sensitive in nature and are used to check the check if an condition is met or not.

For example :– x=True

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

Ans.) The three different types of Boolean operators are :-

1. **and operator ( & )** : This operator returns ‘True’ value if both the operands are true otherwise it returns ‘False’. For example :–

x=True

y=False

print(x and y) # Prints ‘False’

1. **or operator ( | )** : This operator returns ‘True’ value if one of the operands is true or if both are true otherwise it returns ‘False’. For example :-

x=True

y=False

print(x or y) # Prints ‘True’

1. **not operator ( ~ )** : This operator reverses the result, it returns ‘False’ if the statement is ‘True’ and vice-versa. For example :-

x=True

print(not x) #Prints ‘False’

**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. The truth table for each Boolean operator are as follow :-

1. **and operator ( & )**

|  |  |  |
| --- | --- | --- |
| **X-input** | **Y-input** | **and operator output** |
| 0 | 0 | **0** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **1** |

1. **or operator ( | ) :**

|  |  |  |
| --- | --- | --- |
| X - input | Y - input | **or operator output** |
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **1** |

1. **not operator ( ~ ) :**

|  |  |
| --- | --- |
| X - input | **not operator output** |
| 0 | **1** |
| 1 | **0** |

**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?**

Ans.) The six comparison operators are as follow:-

1. **Equal (==)** : Checks whether the values are equal or not and returns a Boolean value after comparison.

Example – (5==5) Returns True

1. **Not Equal (!=) :** Returns a false value when both the operands are equal.

Example - (5!=4) Returns True

1. **Greater than (>) :** Returns a true value when one is greater than other.

Example – (5>4) Returns True

1. **Less than (<) :** Returns a true value when when value is less than other.

Example – (4<5) Returns True

1. **Greater than equal to (>=) :** Returns True when one value is greater or equal to the other.

Example – (5>=4) (5>=5) both returns True

1. **Less than equal to ( <= ):** Returns True when one value is less or equal to the other.

Example – (4<=5) (5<=5) both return True

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

Ans.) The difference between equal and assignment operator is that in assignment operator single-equal (=) symbol is used while in equal to operator double-equal (==) symbol is used.

**For assignment operator:**

x=50

**For equal to operator:**

x==50

**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.) **Block - 1 :**

spam = 0 ( Assignment block )

**Block – 2**

If spam==10: ( Conditional Block )

Print(‘eggs’)

**Block – 3**

if spam > 5: ( Conditional Block )

print('bacon')

**Block – 4**

else: ( else block )

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.

Ans.)

spam=1

if spam==1:

print(‘Hello’)

if spam==2:

print(‘Howdy’)

else:

print(‘Greetings!’)

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

Ans.) CTRL + C ( CONTROL KEY AND C KEY )

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

Ans. The ‘break’ statement immediately leave the loop and ultimately breaks the loop and execute the rest of the code while ‘continue’ statement terminates current iteration and shifts the loop for the next iteration.

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

Ans.)

1. **range(10) :** This function generates a sequence of number 0-9 all inclusive and in for loop it makes the loop iterate 10 times. This sequence starts from 0 by default because no argument for the starting is given.
2. **range(0, 10) :** This function also generates a sequence of number 0-9 all inclusive and in for loop it makes the loop iterate 10 times. This function starts from 0 because the argument of 0 is given for starting the loop.
3. **range(0, 10, 1) :** This function also generates a sequence from 0 to 9 all inclusive but takes 1-extra step after every generation because stepping argument of 1 is passed.

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.)

1. **Using for loop :**

for I in range (1,11):

Print(i)

1. **Using 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?

Ans.) We can call a function from a module by first importing it in the code and then using the syntax **module\_name.function().** For a bacon() function in spam module we will write the following code :-

Import spam

Spam.bacon()