**Python Assignment 2**

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

Boolean values are true and false in python you must write in small case is taken as small case.

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

There are 3 type of Boolean operator are And, or and not operator

* + - && and operator
    - || or operator
    - != 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 evaluates).**

Let us draw a consolidated truth table for all the binary operations, taking the input values as P and Q.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **P** | **Q** | **AND**  **(∧)** | **OR**  **(∨)** | **NAND**  **(~∧)** | **NOR**  **(~∨)** | **Bi-conditional**  **(⇔)** |
| T | T | T | T | F | F | T |
| T | F | F | T | T | F | F |
| F | T | F | T | T | F | F |
| F | F | F | F | T | T | T |

**4. What are the values of the following expressions?**

Expression Values

(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?**

1. < less then
2. >greater then
3. >= greater then equal to
4. <= less then equal to
5. == equal to
6. != Not 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.**

C=9, B=10

C==B checking if C values is equal to B

C=B Assigning B value to C

**7. Identify the three blocks in this code:**

spam = 0

if spam == 10:

#If block it will print eggs when spam ==10

print(“eggs”)

if spam > 5:

#If block it will print bacon when spam has any values which is greater then 5 except 10

print(“bacon”)

else:

#Else block it will print ham spam spam when spam value less then 5

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.**

Spam=0

If spam==1:

Print(“Hello”)

If spam==2:

Print(“Howday”)

Else:

Print(“Greetings)

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

Ctrl+C

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

* **range(10)** produces a sequence of numbers starting from 0 (default start value) and up to, but not including, 10 (default step size of 1). In other words, it returns the sequence [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].
* **range(0, 10)** also produces the same sequence of numbers as above, but explicitly specifies the start value as 0 and the end value as 10.
* **range(0, 10, 1)** produces the same sequence of numbers as above, but explicitly specifies the step size as 1. This is redundant because the default step size is already 1, so

**11.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 I in range(1,11):

Print(I)

I=1

While I<=10:

Print( I)

I+=1

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

from spam import bacon   
bacon()

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

|  |  |  |
| --- | --- | --- |
| **BASIS FOR COMPARISON** | **BREAK** | **CONTINUE** |
| Task | It terminates the execution of remaining iteration of the loop. | It terminates only the current iteration of the loop. |
| Control after break/continue | 'break' resumes the control of the program to the end of loop enclosing that 'break'. | 'continue' resumes the control of the program to the next iteration of that loop enclosing 'continue'. |
| Causes | It causes early termination of loop. | It causes early execution of the next iteration. |
| Continuation | 'break' stops the continuation of loop. | 'continue' do not stops the continuation of loop, it only stops the current iteration. |
| Other uses | 'break' can be used with 'switch', 'label'. | 'continue' can not be executed with 'switch' and 'labels'. |