**Assignment 2 Solutions**

**Ans1** :

There are two values of the boolen data types. (True and False)

We will use Capital T and F and then lowercase .

a**=True**

b**=False**

print(a,type(a))

print(b,type(b))

Output:

True <class 'bool'>

False <class 'bool'>

**Ans2**:

The three different types of Boolean operators in python . (And , Or , Not)

a**=**100

b**=**200

print(a**>**50 **and** b**>**100) #And

print(a**>**200 **or** b**>**100) # Or

print(**not**(a**>**10)) # Not

Output:

True

True

False

**Ans 3:** The Truth tables for the boolean tables are as follows:

* **Truth Table for and operator**  
  True and True is True  
  True and False is False  
  False and True is False  
  False and False is False
* **Truth Table for or operator**  
  True and True is True  
  True and False is True  
  False and True is True  
  False and False is False
* **Truth Table for not operator**  
  True not is False

False not is True

Ans 4:

False

False

True

False

False

True

**Ans5:**

The Six comparison operators available in python are:  
==

 !=

 <

 >

 <=

 =>

**Ans6:**

a**=**3 # Assigning operator that stores value 3 in a variable ‘a’

**if** a**==**3: #comparing values of a variable ‘a’ value and 3

== is the equal to operator that compares two values .

 = is that assignment operator that stores a value in a variable.

**Ans7**: Output:

ham

spam

spam

**Ans 8:**

**def** Sample\_Code(spam):

**if** spam**==**1:

print('Hello')

**elif** spam**==**2:

print('Howdy')

**else**:

print('Greetings')

spamCode(1)

spamCode(2)

spamCode(3)

Hello

Howdy

Greetings

**Ans9:**

Press Ctrl + c to stop a program stuck in an infinite loop.

**Ans10:**

The break statement will move the execution outside the loop .

Whereas the continue/skip the particular  statement and will move the execution to the start of the loop.

**Ans11:** The Differences are :

1. The **range(10)** call range from 0 to 9 (but not include 10)
2. The **range (0,10)**  tells that the loop to start at 0
3. The **range(0,10,1)** tells that the loop to increase the variable by 1 on each iteration

Ans12:

print('Using For Loop')

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

print(i, " ")

print('\n')

print('Using While Loop')

i**=**1

**while** i**<=**10:

print(i, " ")

i**+=**1

Output :

Using For Loop

1 2 3 4 5 6 7 8 9 10

Using While Loop

1 2 3 4 5 6 7 8 9 10

**Ans13:**

This function can be called with spam.bacon().