# Assignment 2
# Solutions

1)

The Boolean data type can have two values “True” or “False”.
1 denotes “True” and 0 denotes “False”.
In Python, we write them as True and False.

2)

The three types of Boolean operators are:

a) AND operator (used as and)
b) OR operator (used as or)
c) NOT operator (used as not)

3)

The Boolean operator truth table is given below:

| **Condition 1** | **Condition 2** | **AND** | **OR** | **NOT Condition 1** |
|---|---|---|---|---|
| False | False | False | False | True |
| False | True | False | True | True |
| True | False | False | True | False |
| True | True | True | True | False |

4)

a) (5 > 4) and (3 == 5)
   False

b) not(5>4)
   False

c) (5 > 4) or (3 == 5)
   True

d) not((5 > 4) or (3 == 5))
   False

e) (True and True) and (True == False)
   False

f) (not False) or (not True)
   True

5)
The six comparisons operators are:

a) ==
b) !=
c) >
d) <
e) >=
f) <=

6)
We use assignment operator while assigning value to a variable for example x = 10
and we use “equal to” operator to check the equality condition for example, x == 20

7)
Here,
the 1st block is of if(spam==10) statement
the 2nd block is of if(spam > 5) statement
the 3rd block is of else statement

8) Code:

```
if(spam==1):
   print("Hello")
elif(spam==2):
   print("Howdy")
else:
   print("Greetings!")
```

9) We will press CTRL + C if our python program get stuck in an endless loop

10) Difference between the break and continue statement:

The break statement is used to exit the loop when a certain condition is met while continue is used to continue while skipping the current iteration. The break statement terminates the loop and the remaining lines of code in that loop are not executed but the continue statement terminates the current iteration only.

11)

a) range(10): It will generate 10 values starting from 0 by default

b) range(0,10): Here, we have passed starting point as 0 and end point as 10. It will generate values starting from 0 till 9 as the end point 10 is not included.

c) range(0,10,1): Here, we have passed another parameter for gap as 1. It will generate 10 values starting from 0 having gap of 1 between them

12)

Using for loop:

```
for i in range(1,11):
    print(i)
```

Using While loop:

```
i=1
while(i<11):
    print(i)
    i = i +1
```

13)

We would call function bacon() from module spam as:

```
Import spam
spam.bacon() #Calling function bacon()
```