Assignment 2(python module)

Ans\_1—the two values of Boolean datatype are

1. True 2) False, the numeric value for True is ‘1’ and False is ‘0’

Ex- if 1==0 will give False and similarly 2==2 will give True

Ans\_2—the three Boolean operators are – or, and, not

Ex- (1==0) or (1==1) will give - True

(1==0) and (1==1) will give- False

not (1==1) will give- False

Ans\_4-- (5 > 4) and (3 == 5) will give— False

not (5 > 4) will give--- False

(5 > 4) or (3 == 5) will give- True

not ((5 > 4) or (3 == 5)) will give-- False

(True and True) and (True == False) will give- False

(not False) or (not True) will give- True

Ans\_3—truth table for

1. Or -

|  |  |  |
| --- | --- | --- |
| Value (0 or 1) | Value (0 or 1) | Output |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

1. And –

|  |  |  |
| --- | --- | --- |
| Value (0 or 1) | Value (0 or 1) | Output |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

1. not –

|  |  |  |
| --- | --- | --- |
| Or - of (1 and 0) | Not(x) |  |
| 1 or 1 | 0 |  |
| 1 or 0 | 0 |  |
| 0 or 1 | 0 |  |
| 0 or 0 | 1 |  |

|  |  |  |
| --- | --- | --- |
| And- of (1 and 0) | Not(x) |  |
| 1 and 1 | 0 |  |
| 1 and 0 | 1 |  |
| 0 and 1 | 1 |  |
| 0 and 0 | 1 |  |

|  |  |  |
| --- | --- | --- |
| Value (1 or 0) | Not(x) |  |
| 1 | 0 |  |
| 0 | 1 |  |
|  |  |  |
|  |  |  |

Ans\_5—six comparison operators are-

|  |  |  |
| --- | --- | --- |
| Name | symbol | Example |
| Equal to | == | 1==1 |
| Greater than | > | 2>1 |
| Less than | < | 3<4 |
| Less than equal to | <= | 2<=4 |
| Greater than equal to | >= | 5>=2 |
| Not equal to | != | 9! =10 |

Ans\_6—the difference b/w equal to (==) and assigning operator (=) is that the first one compares the two values given in left and right if the two values are same than only the code proceeds further, whereas in second case there is nothing like comparison we simply give any value to variable

Ex- 1) when any value given by user is equal to say 5 then only it will give output as

a = int (input ())

if a==5:

print (‘this is correct’)

ex- 2) if I assign a value to any variable like

a=10

the value of variable is 10, there is nothing like comparison

Ans\_9—In case when any program stuck inside infinite loop we should use -CTRL+C, to stop the execution

Ans\_10 break—if we use break inside loop, then if the condition inside loop is correct, then due to break the execution goes out of the loop and executes code which is out of the loop

for i in range (5):

if i==3:

break

print(i)

Continue – if continue is used then the execution goes again at start of the loop when the condition is correct inside loop

for i in range (5):

if i==3:

continue

print(i)

Ans\_11—there is no difference b/w range (10), range (0,10) and range (0,10,1) in terms of output but in first we only gave stop point which is excluded, in second case we are giving start and stop both the points and in last case along with start and stop point we are also giving the step/jump size

But we are getting the same result because by-default the start is set as 0 and also the step/jump as 1

1)for i in range (10):

print(i)

2) for i in range (0,10):

print(i)

3) for i in range (0,10,1):

print(i)

Ans\_12- using for loop-

def test():

for i in range(1,11):

print(i)

using while loop—

def test1():

i=1

while i<11:

print(i)

i=i+1

Ans\_13—if there is module named – spam, and if want to call a function of this module named- bacon () I can do this

Code-----

Import spam as sp

sp. bacon()

just give the input arguments required by the function

Ans\_7—there are 3 else blocks in this

else:

print('ham')

print('spam')

print('spam')

Ans\_8—code is as

def test(a):

if a==1:

print('hello')

elif a==2:

print('howdy')

else:

print('greetings')