**Python – Assignment 2**

|  |  |
| --- | --- |
| 1 | What are the two values of the Boolean data type? How do you write them? |
|  | True and False are the 2 boolean values. Written as ‘**True**’ and ‘**False**’ in python |
| 2 | What are the three different types of Boolean operators? |
|  | ‘and’ – Both the operands should be true, for the expression to return true.  ‘or’ – Both the operands should be false, for the expression to return false.  ‘not’ – Returns the negated Boolean value of the operand. |
| 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 ). |
|  | AND (eg. X and Y)   |  |  |  | | --- | --- | --- | | **X** | **Y** | **X and Y** | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 |   OR (X or Y)   |  |  |  | | --- | --- | --- | | **X** | **Y** | **X or Y** | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 1 |   NOT (not X)   |  |  | | --- | --- | | **X** | **Not X** | | 0 | 1 | | 1 | 0 | |
| 4 | What are the values of the following expressions?  (5 > 4) and (3 == 5)  not (5 > 4)  (5 > 4) or (3 == 5)  not ((5 > 4) or (3 == 5))  (True and True) and (True == False)  (not False) or (not True) |
|  | False  False  True  False  False  True |
| 5 | What are the six comparison operators? |
|  | < : Less Than  <= : Less Than or Equal to  > : Greater Than  >= : Greater Than or Equal To  == : Equal to  != : 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. |
|  | Equal to (==) operator is a conditional operator, used to evaluate whether the LHS operand is equal to the RHS operand; It returns a Boolean value. While, Assignment (=) operator is used to store the RHS value in the target mentioned in LHS. For instance,  We want to check if 3 equals 5, the expression would be: ( 3 == 5 )  We want to store value 3 in the variable ‘val’: ( val = 3 ) |
| 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') |
|  | Block 1:  if spam == 10:  print('eggs')  Block 2:  if spam > 5:  print('bacon')  else:  print('ham')  Block 3:  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. |
|  | if spam == 1:  print(‘Hello’)  elif spam == 2:  print(‘Howdy’)  else:  print(‘Greetings’) |
| 9 | If your programme is stuck in an endless loop, what keys you’ll press? |
|  | Ctrl + C: Exits the kills the running process, and send back control to command line. |
| 10 | How can you tell the difference between break and continue? |
|  | Break: This operation exits the program control flow from the loop iterations to the next statement just after the loop. That’s why it’s said that it breaks out of the loop.  Continue: This operation skips the program control flow to the loop’s next iteration, skipping all the statements in the loop block after this operation. The loop continues from the next iteration. |
| 11 | In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)? |
|  | Range(10): provide numbers in order from 0 to 9. Here it is understood that start is 0 and it would run up to (10 – 1 = 9).  Range(0, 10): provide numbers in order from 0 to 9. Here we explicitly mention the start as 0 and runs up to 9. Range(0, 10, 1): provide numbers in order from 0 to 9, where we explicitly mention the start as 0 and step of 1 and runs up to 9. |
| 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. |
|  | for var in range(1, 10 + 1):  print(var)  var = 1  while var <= 10:  print(var)  var += 1 |
| 13 | If you had a function named bacon() inside a module named spam, how would you call it after importing spam? |
|  | If our import statement looks like follows:   * import spam   Then our function call will be: **spam.bacon()** |