**Question 1**

What do we do to a Python statement that is immediately after an **if** statement to indicate that the statement is to be executed only when the **if** statement is**true**?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Underline all of the conditional code |  |  |  |
| Start the statement with a "#" character |  |  |  |
| Indent the line below the if statement | Correct | 1.00 |  |
| Begin the statement with a curly brace { |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 2**

Which of these operators is **not** a comparison / logical operator?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| = | Correct | 1.00 |  |
| >= |  |  |  |
| != |  |  |  |
| < |  |  |  |
| == |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 3**

What is true about the following code segment:

if x == 5 :

print 'Is 5'

print 'Is Still 5'

print 'Third 5'

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Depending on the value of **x**, either all three of the print statements will execute or none of the statements will execute | Correct | 1.00 |  |
| The string 'Is 5' will always print out regardless of the value for **x**. |  |  |  |
| The string 'Is 5' will never print out regardless of the value for **x**. |  |  |  |
| Only two of the three print statements will print out if the value of **x** is less than zero. |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 4**

When you have multiple lines in an **if** block, how do you indicate the end of the **if** block?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| You use a curly brace { after the last line of the if block |  |  |  |
| You omit the semicolon ; on the last line of the if block |  |  |  |
| You de-indent the next line past the if block to the same level of indent as the original **if** statement | Correct | 1.00 |  |
| You put the colon : character on a line by itself to indicate we are done with the if block |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 5**

You look at the following text:

if x == 6 :

print 'Is 6'

print 'Is Still 6'

print 'Third 6'

It looks perfect but Python is giving you an 'Indentation Error' on the second print statement. What is the most likely reason?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| You have mixed tabs and spaces in the file | Correct | 1.00 |  |
| Python has reached its limit on the largest Python program that can be run |  |  |  |
| Python thinks 'Still' is a mis-spelled word in the string |  |  |  |
| In order to make humans feel inadequate, Python randomly emits 'Indentation Errors' on perfectly good code - after about an hour the error will just go away without any changes to your program |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question Explanation**Please make sure to find the option to auto-expand tabs in your text editor. Or it will be very frustrating when these errors appear in code that \*looks\* perfect.

**Question 6**

What is the Python reserved word that we use in two-way if tests to indicate the block of code that is to be executed if the logical test is false?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| iterate |  |  |  |
| switch |  |  |  |
| else | Correct | 1.00 |  |
| A closing curly brace followed by an open curly brace like this }{ |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 7**

What will the following code print out?

x = 0

if x < 2 :

print 'Small'

elif x < 10 :

print 'Medium'

else :

print 'LARGE'

print 'All done'

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Small All done | Correct | 1.00 |  |
| Medium All done |  |  |  |
| Small Medium LARGE All done |  |  |  |
| All done |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 8**

For the following code,

if x < 2 :

print 'Below 2'

elif x >= 2 :

print 'Two or more'

else :

print 'Something else'

What value of 'x' will cause 'Something else' to print out?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| x = -2 |  |  |  |
| This code will never print 'Something else' regardless of the value for 'x' | Correct | 1.00 |  |
| x = -2.0 |  |  |  |
| x = -22 |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question Explanation**It will never print out because all values for 'x' are either below 2 or greater-than or equal two. So either the **if** or **elif** will print but never the else clause.

**Question 9**

'In the following code (numbers added) - which will be the last line to execute successfully?

(1) astr = 'Hello Bob'

(2) istr = int(astr)

(3) print 'First', istr

(4) astr = '123'

(5) istr = int(astr)

(6) print 'Second', istr

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| 2 |  |  |  |
| 4 |  |  |  |
| 1 | Correct | 1.00 |  |
| 5 |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 10**

For the following code:

astr = 'Hello Bob'

istr = 0

try:

istr = int(astr)

except:

istr = -1

What will the value for **istr after this code executes?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| The **istr** variable will not have a value |  |  |  |
| 9 (the number of characters in 'Hello Bob') |  |  |  |
| -1 | Correct | 1.00 |  |
| It will be a random number depending on the operating system the program runs on |  |  |  |
| Total |  |  |  |