

### 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?

Begin the statement with a curly brace {

Start the statement with a "#" character

Indent the line below the if statement

Underline all of the conditional code

Answer: Indent the line below the if statement

### Question 2

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

=

<

==

>=

>

Answer: =

### Question 3

What is true about the following code segment:

```
if x == 5 :  
    print 'Is 5'  
    print 'Is Still 5'  
    print 'Third 5'
```

Depending on the value of x, either all three of the print statements will execute or none of the statements will execute

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.

Answer: Depending on the value of x, either all three of the print statements will execute or none of the statements will execute

#### Question 4

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

You put the colon : character on a line by itself to indicate we are done with the if block

You use a curly brace { after the last line of the if block

You capitalize the first letter of the line following the end of the if block

Answer: You de-indent the next line past the if block to the same level of indent as the original if statement

#### 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?

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

You have mixed tabs and spaces in the file

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

Answer: You have mixed tabs and spaces in the file

#### 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?

toggle

switch

else

A closing curly brace followed by an open curly brace like this }{

Answer: else

#### 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'
Small
Medium
All done
Small
All done
Small
Medium
LARGE
All done
```

Answer: Small// All done

#### 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?

x = -22

x = -2.0

This code will never print 'Something else' regardless of the value for 'x'

Answer: This code will never print 'Something else' regardless of the value for 'x'

#### 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

5

1

6

2

Answer: 1

#### 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?

false

It will be a random number depending on the operating system the program runs on

-1

9 (the number of characters in 'Hello Bob')

Answer: -1

LATEST SUBMISSION GRADE

100%

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**?

1 / 1

✓ Correct

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

1 / 1

✓ Correct

3. What is true about the following code segment:

1 / 1 point

```
1 if x == 5 :
2     print('Is 5')
3     print('Is Still 5')
4     print('Third 5')
```

✓ Correct

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

1 / 1 point

✓ Correct

5. You look at the following text:

1 / 1 point

```
1 if x == 6 :
2     print('Is 6')
3     print('Is Still 6')
4     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?

✓ Correct

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?

1 / 1 point

✓ Correct

7. What will the following code print out?

1 / 1 point

```
1 x = 0
2 if x < 2 :
3     print('Small')
4 elif x < 10 :
5     print('Medium')
6 else :
7     print('LARGE')
8 print('All done')
```

✓ Correct

8. For the following code,

1 / 1 point

```
1 if x < 2 :
2     print('Below 2')
3 elif x >= 2 :
4     print('Two or more')
5 else :
6     print('Something else')
```

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

✓ Correct

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

1 / 1 point

```
1 (1) astr = 'Hello Bob'
2 (2) istr = int(astr)
3 (3) print('First', istr)
4 (4) astr = '123'
5 (5) istr = int(astr)
6 (6) print('Second', istr)
```

✓ Correct

10. For the following code:

1 / 1 point

```
1 astr = 'Hello Bob'
2 istr = 0
3 try:
4     istr = int(astr)
5 except:
6     istr = -1
```

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

✓ Correct