Chapter 3

TOTAL POINTS 10

1.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
- Underline all of the conditional code
- Indent the line below the if statement

1 point

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

- <=
- >=
- !=
- ==
- =

1 point

3.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.

1 point

4.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 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
- ( ) You omit the semicolon ; on the last line of the if block

1 point

5.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?

- (●) You have mixed tabs and spaces in the file
- ( ) 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

1 point

6.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?

- else
- otherwise
- A closing curly brace followed by an open curly brace like this }{
- iterate

1 point

7.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

LARGE

All done

All done

LARGE

All done

Small

All done

1 point

8.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 = -2.0
- This code will never print 'Something else' regardless of the value for 'x'
- x = -2
- x = 2.0

1 point

9.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)
```

- (•) 1
- ( ) 2
- ( ) 5
- ( ) 4

1 point

10.Question 10
For the following code:

```
astr = 'Hello Bob'
istr = 0
try:
   istr = int(astr)
except:
   istr = -1
```

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

- ( ) It will be the 'Not a number' value (i.e. NaN)
- ( ) The **istr** variable will not have a value
- (•) -1

0

1 point