## Quiz: chapter 1

1. When Python is running in the interactive mode and displaying the chevron prompt (>>>) - what question is Python asking you?

- ( ) What Python script would you like me to run?
- (•) What Python statement would you like me to run?
- ( ) What is the next machine language instruction to run?
- ( ) What is your favourite color?

2. What will the following program print out:

```
>>> x = 15
>>> x = x + 5
>>> print x
```

- (•) 20
- ( ) 5
- ( ) 15
- ( ) "print x"
- ( ) x + 5

3. Python scripts (files) have names that end with:

- ( ) .png
- ( ) .exe
- ( ) .doc
- (•) .py

4. Which of these words are reserved words in Python ?

- [ ] machine
- [x] if
- [x] break
- [ ] todo
- [ ] concat

4. Which of these words is a reserved word in Python ?

- (●) for
- ( ) names
- ( ) pizza
- ( ) payroll

5. What is the proper way to say "good-bye" to Python?

- ( ) #EXIT
- ( ) while
- (●) quit()
- ( ) // stop

6. Which of the parts of a computer actually executes the program instructions?

- ( ) Secondary Memory
- (●) Central Processing Unit
- ( ) Main Memory
- ( ) Input/Output Devices

7. What is "code" in the context of this course?

- ( ) A way to encrypt data during World War II
- ( ) A set of rules that govern the style of programs
- ( ) A password we use to unlock Python features
- (●) A sequence of instructions in a programming language

8. A USB memory stick is an example of which of the following components of computer architecture?

- ○ Output Device
- ○ Central Processing Unit
- ○ Main Memory
- ◉ Secondary Memory

9. What is the best way to think about a "Syntax Error" while programming?

- ◉ The computer did not understand the statement that you entered
- ○ The computer needs to have its software upgraded
- ○ The computer is overheating and just wants you to stop to let it cool down
- ○ The computer has used GPS to find your location and hates everyone from your town

10. Which of the following is **not** one of the programming patterns covered in Chapter 1?

- ○ Conditional Steps
- ○ Sequential Steps
- ○ Repeated Steps
- ◉ Random steps

Week 4

Quiz: chapter 2

1. In the following code,

```
print 98.6
```

What is "98.6"?

- ○ A variable
- ○ A conditional statement
- ○ An iteration / loop statement
- ◉ A constant

2. In the following code,

```
x = 42
```

What is "x"?

- [x] A variable
- [ ] A constant
- [ ] A Central Processing Unit
- [ ] A function

3. Which of the following is a bad Python variable name?

- [ ] SPAM23
- [ ] Spam
- [ ] _spam
- [x] 23spam

4. Which of the following is not a Python reserved word?

- [x] speed
- [ ] else
- [ ] for
- [ ] if

5. Assume the variable x has been initialized to an integer value (e.g., x = 3). What does the following statement do?

```
x = x + 2
```

- [ ] This would fail as it is a syntax error
- [ ] Create a function called "x" and put the value 2 in the function
- [x] Retrieve the current value for x, add two to it and put the sum back into x
- [ ] Produce the value "false" because "x" can never equal "x+2"

6. Which of the following elements of a mathematical expression in Python is evaluated first?

- (●) Parenthesis ( )
- ( ) Multiplication *
- ( ) Addition +
- ( ) Subtraction -

7. What is the value of the following expression

```
42 % 10
```

Hint - the "%" is the remainder operator

- (●) 2
- ( ) 42
- ( ) 4210
- ( ) 420

8. What will be the value of x after the following statement executes:

```
x = 1 + 2 * 3 - 8 / 4
```

- ( ) 8
- ( ) 4
- (●) 5
- ( ) 2

9. What will be the value of x when the following statement is executed:

```
x = int(98.6)
```

- ( ) 100
- (●) 98
- ( ) 99
- ( ) 6

10. What does the Python **raw_input()** function do?

- [x] Pause the program and read data from the user
- [ ] Connect to the network and retrieve a web page.
- [ ] Read the memory of the running program
- [ ] Take a screen shot from an area of the screen

Week 5

Quiz: chapter 3

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**?

- [ ] Underline all of the conditional code
- [ ] Begin the statement with a curly brace {
- [x] Indent the line below the if statement
- [ ] Start the statement with a "#" character

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

- [ ] ==
- [ ] >
- [x] =
- [ ] !=
- [ ] >=

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.

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

- ( ) 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 capitalize the first letter of the line following 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

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
- ( ) 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
- ( ) 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

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
- else
- otherwise
- iterate

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
- Small

  All done
- Small
- LARGE

  All done

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
- x = -22
- x = 2.0
- This code will never print 'Something else' regardless of the value for 'x'

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
- 4
- 2
- 1

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**?

- (●) -1
- ( ) It will be the 'Not a number' value (i.e. NaN)
- ( ) false
- ( ) It will be a random number depending on the operating system the program runs on

Week 6

Quiz: chapter 4

1. Which Python keyword indicates the start of a function definition?

- (●) def
- ( ) continue
- ( ) sweet
- ( ) return

2. In Python, how do you indicate the end of the block of code that makes up the function?

- ( ) You put a # character at the end of the last line of the function
- ( ) You add the matching curly brace that was used to start the function }
- ( ) You put the "END" keyword in column 7 of the line which is to be the last line of the function
- (●) You de-indent a line of code to the same indent level as the **def** keyword

3. In Python what is the **raw_input()** feature best described as?

- ( ) A reserved word
- (•) A built-in function
- ( ) A user-defined function
- ( ) A data structure that can hold multiple values using strings as keys

4. What does the following code print out?

```
def thing():
    print 'Hello'

print 'There'
```

- ( ) Hello
- ( ) Hello

  There
- ( ) thing

  Hello

  There
- (•) There

5. In the following Python code, which of the following is an "argument" to a function?

```
x = 'banana'
y = max(x)
print y
```

- (•) x
- ( ) y
- ( ) print
- ( ) max

6. What will the following Python code print out?

```
def func(x) :
    print x

func(10)
func(20)
```

- x

  20

- func

  func

- x

  10

  x

  20

- 10

  20

7. Which line of the following Python program is useless?

```
def stuff():
    print 'Hello'
    return
    print 'World'

stuff()
```

- print 'World'
- stuff()
- return
- print 'Hello'
- def stuff():

7. Which line of the following Python program is useless?

```
def stuff():
    print 'Hello'
    return
    print 'World'

stuff()
```

- ( ) stuff()
- ( ) print 'Hello'
- (•) print 'World'
- ( ) return
- ( ) def stuff():

8. What will the following Python program print out?

```
def greet(lang):
    if lang == 'es':
        return 'Hola'
    elif lang == 'fr':
        return 'Bonjour'
    else:
        return 'Hello'

print greet('fr'),'Michael'
```

- (•) Bonjour Michael
- ( ) def Michael
- ( ) Hello Michael
- ( ) Hola

  Bonjour

  Hello

9. What does the following Python code print out? (Note that this is a bit of a trick question and the code has what many would consider to be a flaw/bug - so read carefully).

```
def addtwo(a, b):
    added = a + b
    return a

x = addtwo(2, 7)
print x
```

- Traceback
- 7
- 2
- 14

10. What is the most important benefit of writing your own functions?

- Avoiding writing the same non-trivial code more than once in your program
- Following the rule that whenever a program is more than 10 lines you must use a function
- Following the rule that no function can have more than 10 statements in it
- To avoid having more than 10 lines of sequential code without an indent or de-indent

## Week 7

## Quiz: chapter 5

1. What is wrong with this Python loop:

```
n = 5
while n > 0 :
    print n
print 'All done'
```

- This loop will run forever
- The **print 'All done'** statement should be indented four spaces
- There should be no colon on the **while** statement
- **while** is not a Python reserved word

2. What does the **break** statement do?

- ( ) Resets the iteration variable to its initial value
- (•) Exits the currently executing loop
- ( ) Exits the program
- ( ) Jumps to the "top" of the loop and starts the next iteration

3. What does the **continue** statement do?

- ( ) Exits the program
- ( ) Resets the iteration variable to its initial value
- (•) Jumps to the "top" of the loop and starts the next iteration
- ( ) Exits the currently executing loop

4. What does the following Python program print out?

```
tot = 0
for i in [5, 4, 3, 2, 1] :
    tot = tot + 1
print tot
```

- (•) 5
- ( ) 0
- ( ) 15
- ( ) 10

5. What is the *iteration* variable in the following Python code:

```
friends = ['Joseph', 'Glenn', 'Sally']
for friend in friends :
     print 'Happy New Year:',  friend
print 'Done!'
```

- (•) friend
- ( ) Sally
- ( ) Glenn
- ( ) Joseph

6. What is a good description of the following bit of Python code?

```
zork = 0
for thing in [9, 41, 12, 3, 74, 15] :
    zork = zork + thing
print 'After', zork
```

- ○ Find the largest item in a list
- ◉ Sum all the elements of a list
- ○ Find the smallest item in a list
- ○ Count all of the elements in a list

7. What will the following code print out?

```
smallest_so_far = -1
for the_num in [9, 41, 12, 3, 74, 15] :
   if the_num < smallest_so_far :
      smallest_so_far = the_num
print smallest_so_far
```

Hint: This is a trick question and most would say this code has a bug - so read carefully

- ◉ -1
- ○ 74
- ○ 42
- ○ 3

8. What is a good statement to describe the **is** operator as used in the following if statement:

```
if smallest is None :
     smallest = value
```

- ○ Looks up 'None' in the **smallest** variable if it is a string
- ◉ matches both type and value
- ○ Is true if the **smallest** variable has a value of -1
- ○ The if statement is a syntax error

9. Which reserved word indicates the start of an "indefinite" loop in Python?

- def
- indef
- for
- break
- while

10. How many times will the body of the following loop be executed?

```
n = 0
while n > 0 :
    print 'Lather'
    print 'Rinse'
print 'Dry off!'
```

- 0
- This is an infinite loop
- 5
- 1

Assignment

5.2 Write a program that repeatedly prompts a user for integer numbers until the user enters 'done'. Once 'done' is entered, print out the largest and smallest of the numbers. If the user enters anything other than a valid number catch it with a try/except and put out an appropriate message and ignore the number. Enter the numbers from the book for problem 5.1 and Match the desired output as shown.

```
nums = []
while True:
  n = raw_input("Enter a number: ")
  if n == "done":
     print('Invalid input')
     break
  try:
     nums.append(int(n))
  except ValueError:
     print ("Invalid input")

print ("Maximum is %d" % max(nums))
print ("Minimum is %d" % min(nums))
```