EMAT10007 - Introduction to Computer Programming

Class Test: Syntax Test

Overview

- Friday **9th November** is the "Syntax Test" class test. You will be tested on your understanding of the core principles of programming, with emphasis on the specific syntax used by the Python programming language.
- The test will be multiple choice and possibly multiple-answer.
- If you're wondering "where are the answers to the exercises?", look to PyCharm! The best way to revise for this test is to work through the examples in PyCharm, and to modify the examples until you understand the answer to the question.
- Please use the Support Forums to ask questions.

- **Changes and corrections:**

  1. Comments should always confer the intent of the working program, and so these should be used as hints for what the program should do without errors.
  2. Question 6: Line 6 is purposefully written to cause the program to produce an error (that `Number` is a `string`, and not an `int`). The error on the line is that:
     `for Num in range(Number):`
     should be:
     `for Num in range(int(Number)):`
     This is because attempting to produce a `range()` using a `string` produces:
     `TypeError: 'str' object cannot be interpreted as an integer`. I have corrected `range(Num)` to be `range(Number)`.
     The next line, `Sum = Sum + Num`, is not an error because if you correct `range(Number)` to be `range(int(Number))` on the line above, then it would not produce an error.
  3. You should pay close attention to when a program might be missing calls to the conversion functions, such as `int()`, `float()`, and `str()`. This comes from trying to add strings and numbers together, such as in `print()`, or when trying to add two numbers, but one is still a `string`.
  4. New questions have been added. We may test you on things we didn't cover fully before in the Elementary Concepts tests, such as lists, sets, and dictionaries.
  5. Question 9: Line 5 should have been `print(Sentence)`, not `print(WordList)`.

Practice exercises

1. Select the lines that will produce syntax errors:

```
# Ask for a number
```

```
N = input("Enter a number:)

Is it bigger than 10?
if N > 10
    print N
else:
    print(N * 2)
```

☐ Line 1 ☐ Line 2 ☐ Line 4
☐ Line 5 ☐ Line 6 ☐ Line 8

2. Select the missing function:

```
Ingredients = {"Bread":2, "Cheese":3, "Garlic":0.5}
Total = 0

# Calculate the price to make a fondue
for Num in Ingredients._____():
    Total = Total + Num

# Print the total price
print(Total)
```

☐ `key` ☐ `keys` ☐ `value` ☐ `values`

3. Select the lines that will produce syntax errors:

```
Fondue = ["Bread", "Cheese", "Garlic"]
    Total = 0

# Go through the list
for Item in Fondue:
    if len(Item) = 5:
        Total += len(Item)

print(Total)
```

☐ Line 1 ☐ Line 2 ☐ Line 4
☐ Line 5 ☐ Line 6 ☐ Line 7

4. Select the missing operators:

```
Numbers = [5, 6, 4, 4, 3]
Total = 0

for Num in Numbers:
    if Num % 2 == 0:
        Total __ 1
    else:
        Total __ 1

# Print the total
print(Total)
-----
Output: 1
```

☐ -= += ☐ =+ =- ☐ += -=

5. Select the missing operators:

```
Numbers = [5, 6, 4, 4, 3]
Total = 0

for Num in Numbers:
    if Num % 2 == 0:
        Total __ 1
    else:
        Total __ 1

# Print the total
print(Total)
-----
Output: -1
```

☐ -= -= ☐ -= += ☐ += -=

6. Select the lines that will produce syntax errors:

```
# Ask the user to pick a number
Number = input(Pick an integer:)

# Sum all the numbers up to Number
Sum = 0
for Num in range(Number):
    Sum = Sum + Num
```

```
""" Print the sum """
print Sum
```

☐ Line 2 ☐ Line 5 ☐ Line 6
☐ Line 7 ☐ Line 9 ☐ Line 10

7. Select the lines that will produce syntax errors:

```
Word = "hello world"
Vowels = [a,e,i,o,u]

# Print the word, but with each vowel
# converted to upper-case
for Letter in Word:
    if Letter in Vowels
        print(upper(Letter), end="")
    else
        print(Letter, end="")
```

☐ Line 1 ☐ Line 2 ☐ Line 6 ☐ Line 7
☐ Line 8 ☐ Line 9 ☐ Line 10

8. Select the missing functions:

```
WordList = []

# Take a sentence and add each word
# to the WordList
Sentence = "How is the weather?"
for Word in Sentence._______(" "):
    WordList._______(Word)
print(WordList)
-----
Output: ["How", "is", "the", "weather?"]
```

☐ `separate, add` ☐ `split, add` ☐ `split, append` ☐ `separate, append`

9. Select the missing function call:

```
WordList = ["How", "is", "the", "weather?"]

# Combine the words to form a sentence
Sentence = _____________
```

```
print(Sentence)
-----
Output: "How is the weather?"
```

☐ `WordList.join(" ")` ☐ `WordList.combine(" ")`
☐ `join(WordList)` ☐ `" ".join(WordList)`

---

10. Predict the type of A:

```
A = {3, 5, "6", 8.5, 10}
```

☐ `Set` ☐ `Tuple` ☐ `List`

```
A = {"Length" : 1.2, "Width" : 2.5, "Height" : 3}
```

☐ `Set` ☐ `Dict` ☐ `List`

11. Select the correct index:

```
A = [1,2,3,4,5]
print(A[_])
-----
Output: 3
```

☐ `1` ☐ `2` ☐ `3`

```
A = {"0" : 1, "1" : 2, "2" : 3}
print(A[_])
-----
Output: 2
```

☐ `1` ☐ `"1"` ☐ `2` ☐ `"2"`

```
A = {"Length" : 1.2, "Width" : 2.5, "Height" : 3}
```

☐ `Set` ☐ `Dict` ☐ `List`

12. Select the lines that will produce syntax errors:

```
Number = 101
print("Multiples of 10 up to" + Number)
# Print out the multiples of 10
for Num in range(Number):
    if Num % 10 = 0:
        print(Num, "is a multiple of 10!")
```

☐ Line 1 ☐ Line 2 ☐ Line 3
☐ Line 4 ☐ Line 5 ☐ Line 6

13. Select the missing values:

```
WordList = ["Hello", "world"]

# Print the two words as a sentence
print(WordList[_] __ WordList[_])
-----
Output: "Hello world"
```

☐ `1 , 2` ☐ `1 + 2` ☐ `0 + 1` ☐ `0 , 1`

14. Select the missing function:

```
A = {1,2,3,4,5}
A._____(3)
print(A)
-----
Output: {1,2,4,5}
```

☐ `delete` ☐ `remove` ☐ `update` ☐ `clear`

15. Select the lines that will produce syntax errors:

```
import random
Number = input("Please enter an integer:')
Sum = random.randint(1,10)
# Add the user's number to Sum
Sum += Number
```

☐ Line 1 ☐ Line 2 ☐ Line 3 ☐ Line 4
☐ Line 5