COMP10001 Foundations of Computing
Summer Term, 2021

Tutorial Questions: Week 3-1

— VERSION: —, DATE: FEBRUARY 2, 2021 —

Discussion

1. Consider the following while loop and two conversions to for loops. Are the two for loops equivalent? Why might you choose one over the other?

```
count = 0
items = ('eggs', 'spam', 'more eggs')
while count < len(items):
    print(f"we need to buy more {items[count]}")
    count += 1
```

```
items = ('eggs', 'spam', 'more eggs')
for count in range(len(items)):
    print(f"we need to buy more
{items[count]}")
```

```
items = ('eggs', 'spam', 'more eggs')
for item in items:
    print(f"we need to buy more
{item}")
```

Now try Exercises 1 & 2

2. Why is it important to write comments for the code we write? Wouldn't it save time, storage space and processing time to write code without comments?

3. How should we choose variable names? How do good variable names add to the readability of code?

4. What are "magic numbers"? How do we write code without them using global constants?

5. What is a "docstring"? What is its purpose?

   Now try Exercise 3

6. What is a "bug"? What are some debugging strategies which we can use when we find an error?

7. What are the three types of errors we've learned about and what do they mean?

8. How can we use testing to find bugs or confirm our code runs properly? What are some strategies we can adopt to write comprehensive test cases for our code?

   Now try Exercises 4 & 5

Exercises

1. Rewrite the following code using a for loop.

```
i = 2
while i < 6:
    print(f"The square of {i} is {i*i}")
    i = i + 1
```

2. Rewrite the following for loops using while loops.

   (a)
```
colours = ["yellow", "green", "purple"]
for colour in colours[1:]:
    print(colour, "is my favourite colour")
```

(b)
```
MIN_WORD_LEN = 6
long_words = 0
for word in text.split():
    if len(word) >= MIN_WORD_LEN:
        long_words += 1
```

3. Consider the following programs. What are the problematic aspects of their variable names and use of magic numbers? What improvements would you make to improve readability?

(a)
```
a = float(input("Enter days: "))
b = a * 24
c = b * 60
d = c * 60
print("There are", b, "hours,", c, "minutes", d, "seconds in", a, "days")
```

(b)
```
word = input("Enter text: ")
words = 0
vowels = 0
word_2 = word.split()
for word_3 in word_2:
    words += 1
    for word_4 in word_3:
        word_5 = word_4.lower()
        if word_5 in "aeiou":
            vowels += 1
if vowels/words > 0.4:
    print("Above threshold")
```

4. Find the errors in the following programs, classifying them as (a) syntax, (b) runtime or (c) logic errors. Fix them with a correct line of code.

(a)
```
def disemvowel(text):
    """ Returns string `text` with all vowels removed """
    vowels = ('a', 'e', 'i', 'o', 'u')
    answer = text[0]
    for char in text:
        if char is not in vowels:
            answer = char + answer
    print(answer)
```

(b)
```
def big-ratio(nums, n):
    """ Calculates and returns the ratio of numbers
    in list `nums` which are larger than `n` """
    n = 0
    greater_n = 0
    for number in nums:
        if number > n:
            greater_n += 1
            total += 1
    return greater_n / total

nums = [4, 5, 6]
    low = 4
print(f"{100*big_ratio(nums, low)}% of numbers are greater than {low}")
```

5. Imagine we are given some code which we must test to ensure it works as intended. The code takes a student's ID and a year as input, fetches their records and calculates their average mark for that academic year. Describe three test cases you could construct to test different aspects of the code's functionality.

Problems

1. Write a function which takes a string containing an FM radio frequency and returns whether it is a valid frequency. A valid frequency is within the range 88.0-108.0 inclusive with 0.1 increments, meaning it must have only one decimal place.

2. Write a function which takes two lists as input and returns a list containing the numbers which they both have in common.