

Week 4 Tutorial



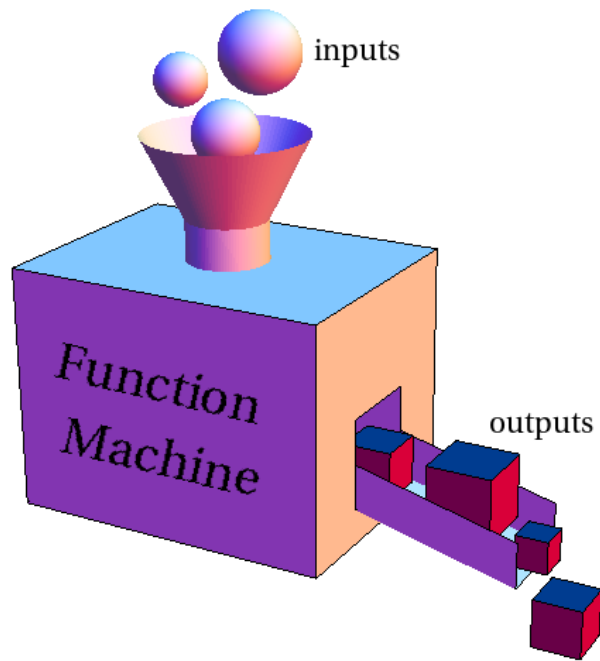
COMP10001 – Foundations of Computing

Semester 1, 2025

Clement Chau

- Conditionals and Functions
- Methods, comments, and tuples

Revision: Functions!



```
sem1-2025 > week-4 > functions.py > ...
```

```
1  # This is an example of a function in Python
2  def add(a, b):
3      return a + b
4
5  result = add(1, 2)
6  print(result) # result = 3
```

Exercise 1, what does this program produce when executed?

```
def ave(a, b):  
    result = (a + b) / 2
```

```
def ave_p(a, b):  
    result = (a + b) / 2  
    print("p", result)
```

```
def ave_r(a, b):  
    result = (a + b) / 2  
    return result
```

```
def ave_pr(a, b):  
    result = (a + b) / 2  
    print("pr", result)  
    return result
```

```
res = ave(1, 2) → None
```

```
res_p = ave_p(1, 2) → None
```

```
res_r = ave_r(1, 2) → 1.5
```

```
res_pr = ave_pr(1, 2) → 1.5
```

print?

4 Values?

P ~~1.5~~

pr 1.5

Exercise 2, what's wrong with this code?

```
def calc(n1, n2):  
    answer = n1 + (n1 * n2)  
    print(answer) return
```

return ?

```
num = int(input("Enter the second number: "))  
result = calc(2, num)  
print("The result is:", result)
```

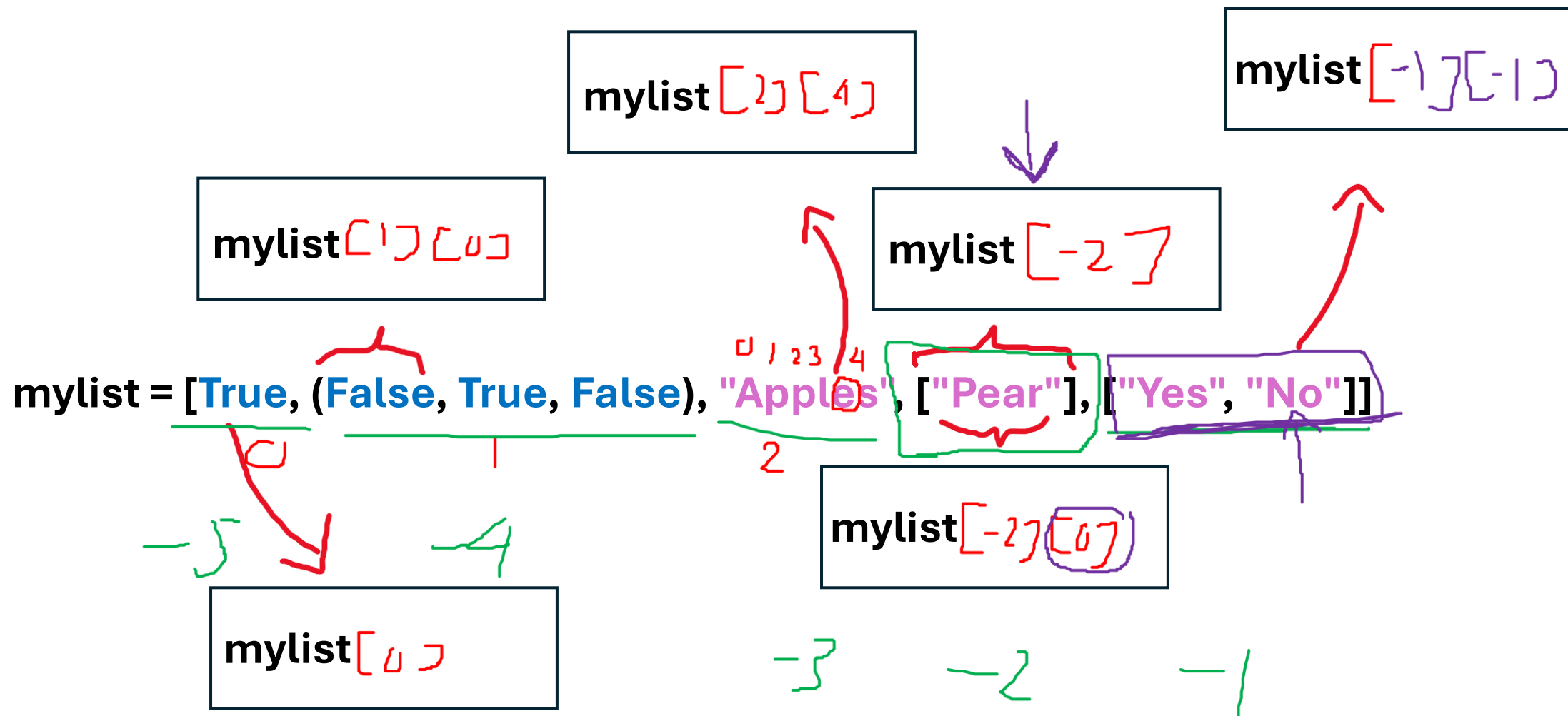
None

$s = \text{"hi"} \rightarrow s.strip()$ ↓ ↓

Exercise 3, given that $s = \text{"Computing is FUN!"}$,

s.isupper() False	s.upper() "COMPUTING IS FUN!"	s.endswith("FUN!") True	s.count('i') 2
<u>s.strip('!')</u> "Computing is FUN"	s.replace('i', '!') "Comput!ng ! FUN!"	s.split() ["Computing", "is", "FUN!"]	s.isdigit() False

Revision: Indexing → str, list, tuple

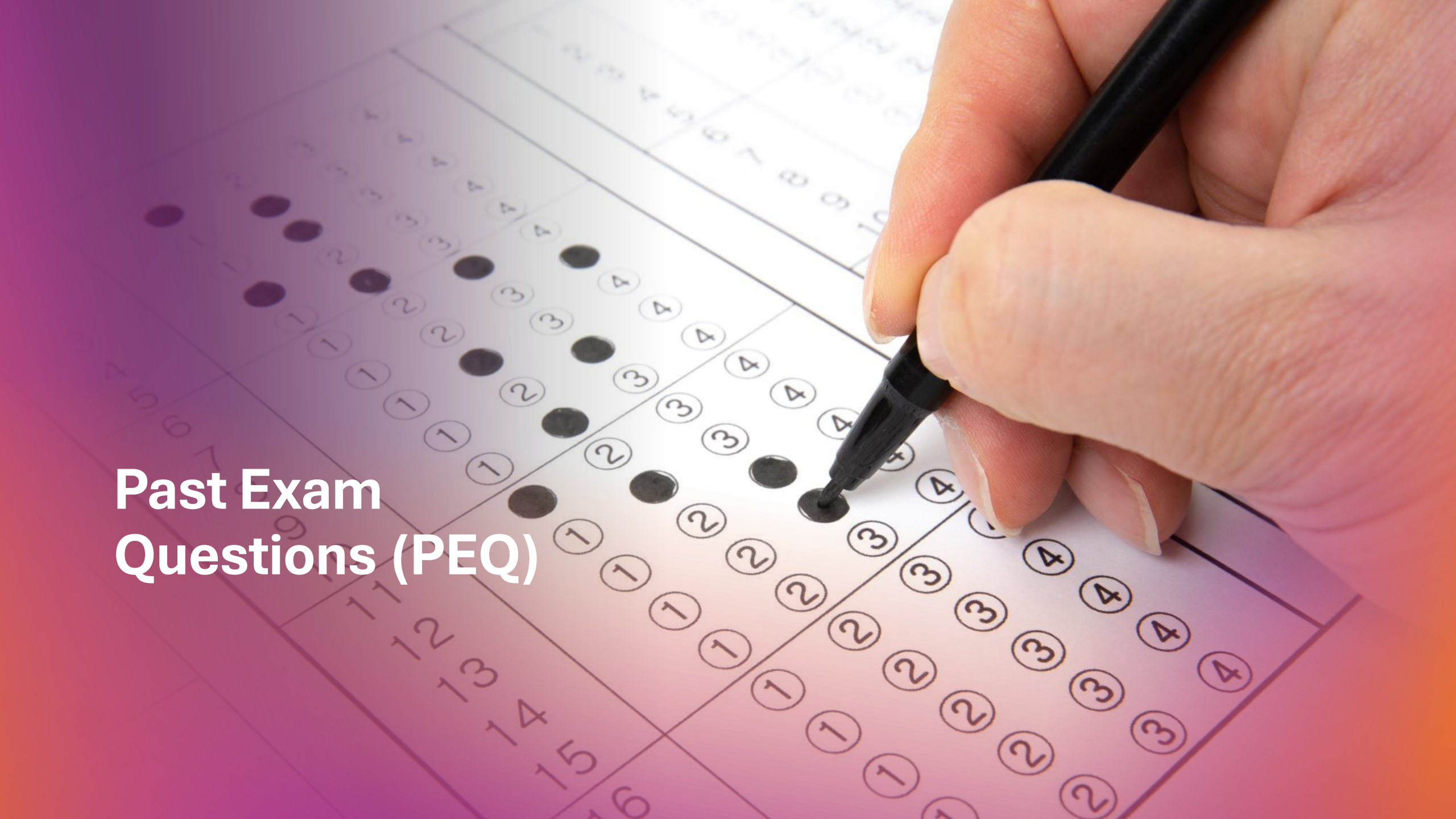


Exercise 4, given that `lst = [2, ("green", "eggs", "ham"), False]`:

Handwritten annotations above the list:
- Above `2`: `[]`
- Above `"green"`: `\`
- Above `"eggs"`: `1`
- Above `"ham"`: `2`
- Below `2`: `-3`
- Below `"eggs"`: `-2`
- Below `False`: `-1`

<code>lst[2]</code> <i>False</i>	<code>lst[1][-2]</code> <i>"eggs"</i>	<code>lst[1][-2][:3]</code> <i>"eggs"[0:3]</i> <i>→ "egg"</i>
<code>lst.append(5)</code> <code>print(lst)</code> <i>Adds 5 to the list</i>	<code>lst.pop(2)</code> <code>print(lst)</code> <i>False ;</i> <i>[2, ("green", "eggs", "ham")]</i>	<code>lst.reverse();</code> <code>print(lst)</code> <i>Reverse the list</i> <i>lst[::-1].reverse()</i>

Past Exam Questions (PEQ)

A close-up photograph of a hand holding a black pen, filling out a multiple-choice exam bubble sheet. The hand is positioned over a grid of bubbles, with the pen tip touching one of the bubbles. The grid is organized into columns, with numbers 1, 2, 3, and 4 printed in circles next to each column of bubbles. The background is a soft, out-of-focus purple and pink gradient.

`list('hello')`

`('h,')`

PEQ 1, what does the following code produce as output?

<p><code>22 % 4 * 4 + 1.5 - 22 // 3</code></p> <p>Handwritten calculation:</p> <p>$22 \% 4 = 2$ $2 * 4 = 8$ $8 + 1.5 = 9.5$ $22 // 3 = 7$ $9.5 - 7 = 2.5$</p>	<p><code>"hello"[:3] + "p"</code></p> <p>Handwritten output: <code>"help"</code></p>
<p><code>(1, 2) + (3) + (4, 5)</code></p> <p>Handwritten: <code>TypeError</code></p> <p>Handwritten: <code>(3,)</code></p>	<p><code>tuple('abc') + (4,)</code> $(4) \rightarrow 4$</p> <p>Handwritten: <code>('a', 'b', 'c') + (4,)</code> <code>= ('a', 'b', 'c', 4)</code></p>

PEQ 2a, one liners only ⚠

- (a) Suppose that str1 and str2 are two strings, and that k is a positive integer. Give a single Python assignment statement that assigns True to match if str1 and str2 have the same first k characters, and assigns False to match if not.

A: `match = str1[:k] == str2[:k]`

PEQ 2b, one liners only ⚠

- (b) Suppose that `lst` is a non-empty list of numbers. Give a single Python assignment statement that assigns the difference between the largest and smallest numbers in `lst` to the variable `diff`.

A: `diff = max(lst) - min(lst)`

PEQ 2c, one liners only ⚠

- (c) Suppose that `text` is a Python string. Give a single Python assignment statement that assigns the number of words in text to `wrds`, where a “word” is any non-blank sequence of characters. (Hint: A method covered in previous exercises may be useful).

A: `wrds = len(text.split())`

Programming on Paper



Problem 1

1. Write a function which converts a temperature between degrees Celsius and Fahrenheit. It should take a float, the temperature to convert, and a string, either 'c' or 'f' indicating a conversion from degrees Celsius and Fahrenheit respectively. The formulae for conversion is: $\text{Fahrenheit} = \text{Celsius} * 1.8 + 32$

Answer:

A:

```
def convert_temp(degrees, unit):  
    if unit == 'c':  
        result = (degrees * 1.8) + 32  
        return result  
    elif unit == 'f':  
        result = (degrees - 32) / 1.8  
        return result
```

return result

Problem 2

2. Write a function which takes a sentence as a single argument (in the form of a string), and evaluates whether it is valid based on whether the first letter is capitalised and the last character is a full stop. Return a Boolean value True or False.

Answer:

A:

```
def check(sentence):  
    if "A" <= sentence[0] <= "Z" and sentence[-1] == ".":  
        return True  
    else:  
        return False
```

or

```
def check(sentence):  
    return "A" <= sentence[0] <= "Z" and sentence[-1] == "."
```

or

```
def check(sentence):  
    return sentence[0].isupper() and sentence[-1] == "."
```

Problem 3

3. Simple loop. Write two programs to print integer numbers from -5 to 5 (inclusive). The first program should use a for loop and the second program should use a while loop.

Answer:

A: *For loop:*

```
for i in range(-5, 6):  
    print(i)
```

While loop:

```
i = -5  
while i <= 5:  
    print(i)  
    i += 1
```


Problem 4

4. **Challenge:** 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. `valid_fm('103.14')` should return `False`.

Answer:

A:

```
def valid_fm(freq):  
    """ Takes an FM frequency as a string and returns a boolean  
    value indicating whether it's a valid frequency or not. """  
  
    # Checks length of string  
    if len(freq) != 4 and len(freq) != 5:  
        return False  
  
    # Checks characters conform to a number  
    if not freq[:-2].isdigit() or freq[-2] != "." or not freq[-1].isdigit():  
        return False  
  
    # Avoid preceding 0 (e.g. "088.2" is not valid)  
    if freq[0] == "0":  
        return False  
  
    # Returns based on final range check  
    return 88.0 <= float(freq) <= 108.0
```

Independent Work

- **Do worksheets 6, 7, 8** on Ed (**due next Monday at 6pm**)
 - Remember that **Ed worksheets contributes to 10% of your total score!**
- **Raise your hand** if you have any questions!

Scan here for annotated slides

