Pytest practice

Practice #1

We want to write a function `obscure_text` that:

- takes in a string
- returns the string with some letters replaced by numbers
- note that this is a common but very inefficient password practice

The character substitutions are the following:

- `o` and `O` become `0`
- `e` and `E` become `3`
- `I` becomes `1`
- `l` becomes `|`
- `a` becomes `@`

For example: `Hello, my name is Tim` => `H3ll0, my n@m3 is Tim`.

Write the code and the tests for the function `obscure_text`. Practice with tests in the same file as your code, as well as in a separate file (using `import`).

Practice #2

Create a new Python file, and use the following variable as a list of courses that may or may not be available at BCIT.

```
COURSES = ["ACIT2515", "ACIT1620", "ACIT0220", "ACIT0320",
"ACIT1420", "ACIT1951", "COMM0330", "COMM0350", "COMM1650",
"ENGL0220", "ENGL0510", "MATH0150", "ENGL0450", "PROJ0900",
"PROJ1360"]
```

Write a function `department_courses(course_list, department)` that takes in a list of courses (see above) and a department name (for example `ENGL`) and returns a list of all courses in that department (first 4 letters). Your code should pass the following test:

```python
def test_department_courses():
    assert department_courses([], "") == []
    assert department_courses(["ACIT1234"], "") == []
    assert department_courses(["ACIT1234"], "ACI") == []
    assert department_courses(["ACIT1234"], "ENGL") == []
    assert department_courses(["ACIT1234"], "ACIT") == ["ACIT1234"]
    assert department_courses(["ACIT1234", "ACIT5678"], "ACIT") == ["ACIT1234", "ACIT5678"]
    assert department_courses(["ACIT1234", "ACIT5678", "ENGL1234"], "ACIT") == ["ACIT1234", "ACIT5678"]
    assert department_courses(["ACIT1234", "ACIT5678", "ENGL1234"], "ENGL") == ["ENGL1234"]
```

Test-Driven Development: make changes / add features

For each of the changes below, adjust your code **AND** your tests to make it work.

- We want the function `department_courses` to work regardless of the case.
- `department_courses` should return `[]` if the first argument is not a list or a tuple.
- `department_courses` should raise an `AttributeError` if the second argument is not a string with 4 letters.