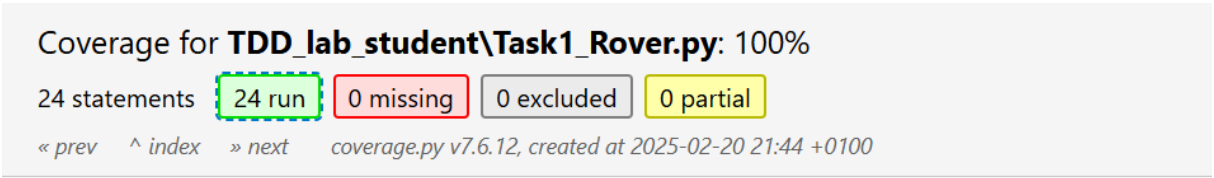
Test-Driven Development Lab

André Plancha
andre.plancha@hotmail.com
MALMÖ UNIVERTITY
Yana Zlatanova
yana-zlatanova-gold@gmail.com

February 21, 2025

Answer the following questions for Task 1:

1. **Please run the test coverage module, generate the HTML report, and attach the screenshot of test coverage (see the lecture for reference)**




```
1  class rovar(object):
2    _LOWER_CONSTANTS = "bcdfghjklmnpqrstvwxz"
3    _UPPER_CONSTANTS = "BCFGHJKLDMNPQRSTVWXZ"
4
5    def enrove(self, normal: str) -> str:
6      """Encode the string in rovarspraket.
7      Args:
8          normal (str): Normal string
9      Returns:
0          (str) Encoded String
1      """
2      if normal is None:
3        return None
4
5      builder = ""
6      for c in normal:
7        if c in self._LOWER_CONSTANTS:
8          builder += c + "o" + c
9        elif c in self._UPPER_CONSTANTS:
0          builder += c + "O" + c
1        else:
2          builder += c
3
4      return builder
5
6    def derove(self, rov: str) -> str:
7      """
```

2. What types of coverage were measured?

Line Coverage: The percentage of code lines executed.

3. What is the code coverage you achieved with the test cases?

The coverage is 100 %.

4. Please state and describe the defect found in the code. (There is a bug!).

The defect in the code is an inconsistency in the CONSTANTS reference strings where 'g' is missing from *LOWER_CONSTANTS and 'D' is missing from* UPPER_CONSTANTS, causing these letters to be treated not as CONSTANTS but as vowels (or other character) and remain unchanged during encoding/decoding which is incorrect.

5. Which test case (s) found the defect (copy and paste the test case to answer the question)?

The defect was found in the test case for encoding and decoding of all CONSTANTS in lower and upper case in a string.

```
# Test encoding
  def test_consonants(self):
        test_cases = {
            # Lowercase
            "b": "bob",
            "c": "coc",
            "d": "dod",
            "f": "fof",
            "g": "gog",
            "h": "hoh",
```

```
            "z": "zoz",
            # Multiple consonants
            "br": "bobror",
            "str": "sostotror",
            # Mixed case consonants
            "B": "BOB",
            "Kr": "KOKror"
        }

        for input_str, expected in test_cases.items():
            encoded = self.rv.enrove(input_str)
            self.assertEqual(encoded, expected,
                f"Encoding failed for '{input_str}': got '{encoded}', expected '{expected}'")
            # Test decoded
            self.assertEqual(self.rv.derove(encoded), input_str)
```

6. How did you fix the defect(s) found in the code?

The bug was mitigated by adding "g" and "D" to the string with reference constants.

Answer the following ques6ons for Task 2:

1. **Please run the test coverage module, generate the HTML report, and attached the screenshot of test coverage (see the lecture for reference)**

2. What types of coverage were measured?

Line Coverage: The percentage of code lines executed

3. How many test cases were written to test the user stories mentioned in task 2?

A total of 16 tests we're written, some with a few subtests. Most of them we're one test (with subtests) per story.

4. What is the code coverage you achieved with the test cases?

We achieved 100% code coverage.