

# Software Engineering Lab 6

Name: Rithvik Rajesh Matta

SRN: PES2UG23CS485

Sec: H

```
Keyboard input: apt
● PS C:\Users\rithv\sem5\SE\PES2UG23CS485> python processor.py
==== Running sanitize_string ====
Enter a string with special characters (!,@,#,$,%): hi!!!
Sanitized String: hi

==== Running parse_int_list ====
Enter a CSV of integers (e.g. 1,2,3,4): 12345, 12345,5432, 536464, 434
Parsed Integer List: [12345, 12345, 5432, 536464, 434]

==== Running reverse_words ====
Enter a sentence without punctuation: hi my name is rithvik
Reversed Words Sentence: ih ym eman si kivhtir
○ PS C:\Users\rithv\sem5\SE\PES2UG23CS485> █
```

Ss2

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> pytest test_processor.py
=====
platform win32 -- Python 3.7.0, pytest-7.4.4, pluggy-1.2.0
rootdir: C:\Users\rithv\sem5\SE\PES2UG23CS485
plugins: hypothesis-6.79.4, cov-4.1.0
collected 3 items

test_processor.py FFF

=====
test_SANITIZE_STRING_no_crash

s = None

@given(st.text() | st.none())
def test_SANITIZE_STRING_no_crash(s):
    try:
        result = sanitize_string(s)

test_processor.py:8:
-----
data = None

def sanitize_string(data):
    """
    Removes special characters and trims the input.
    Assumes data is a non-empty string.
    """
    data = data.strip()
E     AttributeError: 'NoneType' object has no attribute 'strip'

processor.py:6: AttributeError
During handling of the above exception, another exception occurred:

@given(st.text() | st.none())
> def test_SANITIZE_STRING_no_crash(s):

test_processor.py:6:
```

Ln 34, Col 1 Spaces: 4 UTF-8 CRLF {} Python

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> pytest test_processor.py
test_processor.py:6:
-----
s = None

@given(st.text() | st.none())
def test_sanitize_string_no_crash(s):
    try:
        result = sanitize_string(s)
    except Exception as e:
        assert False, f"sanitize_string crashed with input={s!r}, error={e}"
E         AssertionError: sanitize_string crashed with input=None, error='NoneType' object has no attribute 'strip'
E         assert False
E         Falsifying example: test_sanitize_string_no_crash(
E             s=None,
E         )

test_processor.py:10: AssertionERROR
-----
```

```
s = None

@given(st.text() | st.none())
def test_parse_int_list_safe(s):
    try:
        result = parse_int_list(s)
    except Exception as e:
        assert False, f"parse_int_list crashed with input={s!r}, error={e}"
E         AssertionError: parse_int_list crashed with input=None, error='NoneType' object has no attribute 'split'
E         assert False
E         Falsifying example: test_parse_int_list_safe(
E             s=None,
E         )
```

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> pytest test_processor.py
processor.py:16: AttributeError
During handling of the above exception, another exception occurred:

@given(st.text() | st.none())
>   def test_parse_int_list_safe(s):
test_processor.py:16:

s = None

@given(st.text() | st.none())
def test_parse_int_list_safe(s):
    try:
        result = parse_int_list(s)
    except Exception as e:
        assert False, f"parse_int_list crashed with input={s!r}, error={e}"
E         AssertionError: parse_int_list crashed with input=None, error='NoneType' object has no attribute 'split'
E         assert False
E         Falsifying example: test_parse_int_list_safe(
E             s=None,
E         )
```

```
test_processor.py:20: AssertionERROR
-----
```

```
s = None

@given(st.text() | st.none())
def test_reverse_words_safe(s):
    try:
        result = reverse_words(s)
    except Exception as e:
        assert False, f"reverse_words crashed with input={s!r}, error={e}"
E         AssertionError: reverse_words crashed with input=None, error='NoneType' object has no attribute 'split'
E         assert False
E         Falsifying example: test_reverse_words_safe(
E             s=None,
E         )
```

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> pytest test_processor.py
Assumes sentence is non-empty and contains no punctuation.
"""
>     words = sentence.split()
E     AttributeError: 'NoneType' object has no attribute 'split'

processor.py:24: AttributeError

During handling of the above exception, another exception occurred:

@given(st.text() | st.none())
>     def test_reverse_words_safe(s):

test_processor.py:28:

s = None

@given(st.text() | st.none())
def test_reverse_words_safe(s):
    try:
        result = reverse_words(s)
    except Exception as e:
        assert False, f"reverse_words crashed with input={s!r}, error={e}"
E     AssertionError: reverse_words crashed with input=None, error='NoneType' object has no attribute 'split'
E     assert False
E     Falsifying example: test_reverse_words_safe(
E         s=None,
E     )

test_processor.py:32: AssertionError
=====
short test summary info
FAILED test_processor.py::test_SANITIZE_string_no_crash - AssertionError: sanitize_string crashed with input=None, error='NoneType' object has no attribute 'strip'
FAILED test_processor.py::test_PARSE_int_list_safe - AssertionError: parse_int_list crashed with input=None, error='NoneType' object has no attribute 'split'
FAILED test_processor.py::test_reverse_words_safe - AssertionError: reverse_words crashed with input=None, error='NoneType' object has no attribute 'split'
=====
3 failed in 1.25s
```

### Ss3

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> python processor.py
===== Running sanitize_string =====
Enter a string with special characters (!,@,#,$,%): @hello world!!!
Sanitized String: hello world

===== Running parse_int_list =====
Enter a CSV of integers (e.g. 1,2,3,4): 12345,6543,2345,987,65432
Parsed Integer List: [12345, 6543, 2345, 987, 65432]

===== Running reverse_words =====
Enter a sentence without punctuation: i want to be placed early
Reversed Words Sentence: i tnaw ot eb decalp ylrae
```

### Ss4

```
PS C:\Users\rithv\sem5\SE\PES2UG23CS485> pytest test_processor.py
=====
test session starts
platform win32 -- Python 3.7.0, pytest-7.4.4, pluggy-1.2.0
rootdir: C:\Users\rithv\sem5\SE\PES2UG23CS485
plugins: hypothesis-6.79.4, cov-4.1.0
collected 3 items

test_processor.py ... [100%]

=====
3 passed in 1.42s
```

## *Reflections*

### *1. How did Hypothesis help?*

*Hypothesis automatically produced diverse and unpredictable test cases — including edge conditions like `None`, blank strings, and invalid or non-numeric data. This uncovered issues that wouldn't normally appear in traditional unit tests, improving the overall reliability and robustness of the functions.*

### *2. Why use Fuzzing in CI/CD Pipelines?*

*Adding fuzz testing to a CI/CD workflow ensures that every code update is automatically evaluated with random and potentially problematic inputs. This helps detect hidden vulnerabilities or crashes early in development, maintaining code stability and preventing regressions before deployment.*