

```

1 import pytest
2 from classes_def import Text, Sent, LexEntry
3
4 def test_sentences():
5     text = Text("Hello, this is an example text. It
6     has multiple sentences.")
7     sentences = text.split_into_sent()
8     sent = Sent(sentences)
9
10    print("Sentences:")
11    for i in range(len(sentences)):
12        try:
13            print(f"Sentence {i}: {sent.get_sentence(
14                i)}")
15        except IndexError as e:
16            print(e)
17
18    assert sentences == ["Hello, this is an example
19    text.",
20                        "It has multiple sentences."
21    ], f"Expected sentences did not match: {sentences}"
22
23 def test_lex_entries():
24     text = Text("Hello, this is an example text. It
25     has multiple sentences.")
26     sentences = text.split_into_sent()
27     sent_chunk = Sent(sentences)
28     for sentence in sentences:
29         lex_entry = LexEntry(sentence)
30         words = lex_entry.split_into_words()
31         print(f"\nWords in '{sentence}':")
32         for word in words:
33             print(f"Word: {word}, Length: {len(word)}
34             , Index: {lex_entry.get_word_index(word)}")
35
36     first_sentence_lex = LexEntry(sentences[0])
37     assert first_sentence_lex.split_into_words() == [
38         "hello", "this", "is", "an", "example",
39         "text"], f"Expected words did not match: {

```

```

33 first_sentence_lex.split_into_words()}"
34
35     second_sentence_lex = LexEntry(sentences[1])
36     assert second_sentence_lex.split_into_words
37     () == ["it", "has", "multiple",
38           "sentences"], f"Expected words did not match: {
39 second_sentence_lex.split_into_words()}"
40
41 def test_lemma():
42     text = Text("Hello, this is an example text. It
43 has multiple sentences.")
44     sentences = text.split_into_sent()
45     sent = Sent(sentences)
46
47     for sentence in sentences:
48         lex_entry = LexEntry(sentence)
49         words = lex_entry.split_into_words()
50
51         for word in words:
52             pos = lex_entry.get_pos(word)
53             lemma = lex_entry.lemmatize_word(word)
54
55             # Print or assert as needed for testing
56             print(f"Word: {word}, POS: {pos}, Lemma:
57 {lemma}")
58
59             # Example assertion (adjust as needed
60             # based on your expected outcomes)
61             # This example assumes that 'example'
62             # should be lemmatized to 'example'
63             if word == 'example':
64                 assert lemma == 'example', f"Expected
65 'example', but got '{lemma}'"
66             elif word == 'has':
67                 assert lemma == 'have', f"Expected '
68 have', but got '{lemma}'"
69
70 if __name__ == "__main__":
71     pytest.main()

```