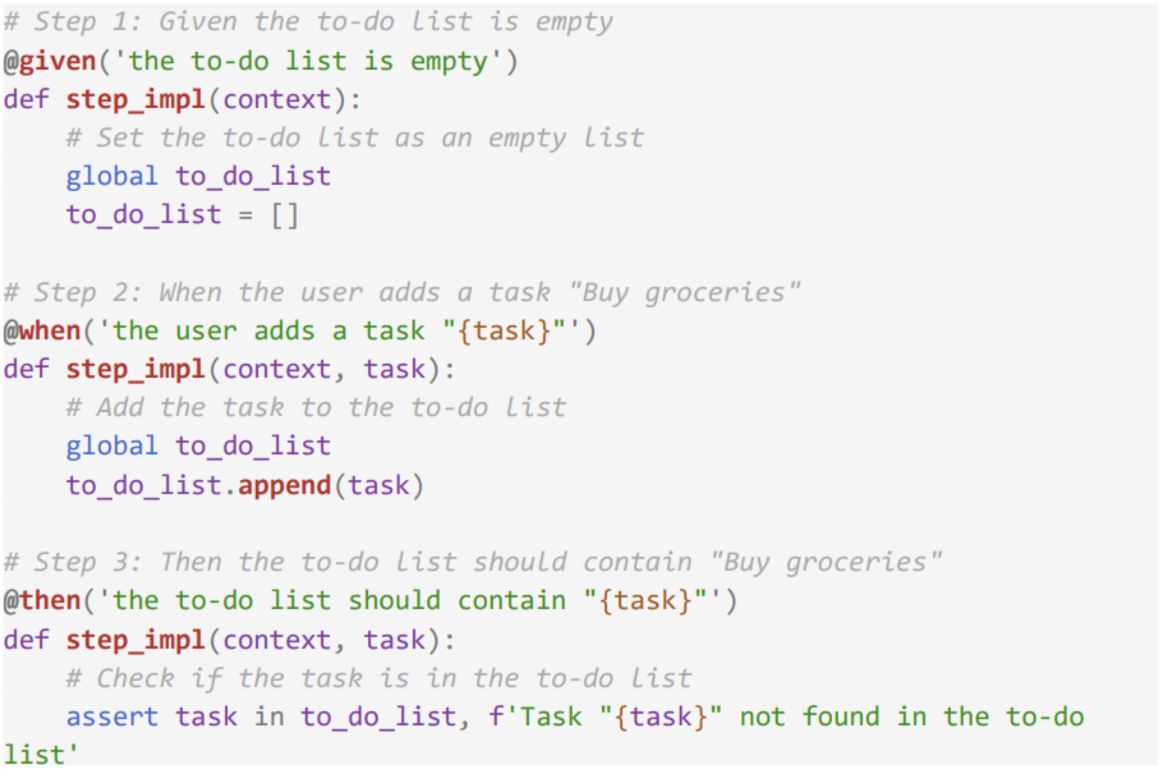
Name: Freddy Tenesaca

Course: SI II P101

Acceptance Testing



```
# Step 1: Given the to-do list is empty
@given('the to-do list is empty')
def step_impl(context):
    # Set the to-do list as an empty list
    global to_do_list
    to_do_list = []

# Step 2: When the user adds a task "Buy groceries"
@when('the user adds a task "{task}"')
def step_impl(context, task):
    # Add the task to the to-do list
    global to_do_list
    to_do_list.append(task)

# Step 3: Then the to-do list should contain "Buy groceries"
@then('the to-do list should contain "{task}"')
def step_impl(context, task):
    # Check if the task is in the to-do list
    assert task in to_do_list, f'Task "{task}" not found in the to-do
list'
```

Update Task Details:

Allowing users to update the details of a specific task. This could include modifying the task name, status, or description.

Example Scenario:

Scenario: Update task details in the to-do list

Given the to-do list contains tasks:

Task	Status	Description
Buy groceries	Pending	Go to the supermarket and buy food

When the user updates the task "Buy groceries" with new details:

New Task Name	New Status	New Description
Buy weekly groceries	In Progress	Visit the supermarket and purchase essentials

Then the to-do list should reflect the updated details:

Task	Status	Description
Buy weekly groceries	In Progress	Visit the supermarket and purchase essentials

Prioritize Tasks:

Introducing the ability to prioritize tasks to help users focus on what needs to be done first.

Example Scenario:

Scenario: Prioritize tasks in the to-do list

Given the to-do list contains tasks:

Task	Status	Description
Pay bills	Pending	Ensure all utility bills are paid
Work on project	Pending	Complete project tasks by the deadline

When the user prioritizes tasks with the following order:

Task
Work on project
Pay bills

Then the to-do list should reflect the prioritized order:

Task	Status	Description
Work on project	Pending	Complete project tasks by the deadline

Pay bills	Pending	Ensure all utility bills are paid

Two additional scenarios for your To-Do List Manager:

Scenario: Add a task with a description to the to-do list

Given the to-do list is empty

When the user adds a task "Buy groceries" with description "Go to the supermarket and buy fruits and vegetables"

Then the to-do list should contain:

Task	Description
Buy groceries	Go to the supermarket and buy fruits and vegetables

Scenario: Remove a task from the to-do list

Given the to-do list contains tasks:

Task
Buy groceries
Pay bills

When the user removes the task "Buy groceries"

Then the to-do list should contain only the task "Pay bills"

Scenario: Mark a task as completed

Given the to-do list contains tasks:

| Task | Status |

| Buy groceries | Pending |

When the user marks task "Buy groceries" as completed

Then the to-do list should show task "Buy groceries" as completed

Scenario: Clear the entire to-do list

Given the to-do list contains tasks:

| Task |

| Buy groceries |

| Pay bills |

When the user clears the to-do list

Then the to-do list should be empty