Quiz 2 - Computational Physics II

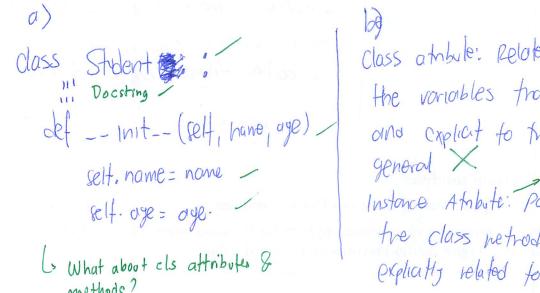
NAME: Juan Doniel Vosconez Vela SC	ORE:
Date: Thursday 20 March 2025 Duration: 45 minutes	(6)(1)
Credits: 20 points (4 questions) Type of evaluation: LAB	120

Provide short and concise answers to the following items. Code syntax should be clear.

1. (5 points) Python classes

(a) Provide a simple code snippet of a Python class showing how it is defined and its components.

-9.2 (b) Explain the difference between an instance attribute and a class attribute in a Python class.



class athbule: Related to self, are
the variables that reffers directly
and explicit to the class in
general X
Instance Athbute: Parameters inside
the class netrods and not
explicitly related to the class factory.

2. (5 points) Python decorators

(a) What is a decorator in Python, and what is its purpose?

(b) Provide a simple code example of a decorator and explain what it does.

def mysecoroter (first) / Docstvings A decoroter is a function that

def unapper (first) / an missing adds functionality to another class

first ()

print ("Extra finetanolis") / class wide.

Comy decoroter /

class wide.

function

Comy decoroter /

class wide.

3. (5 points) Python packages

- (a) Describe the typical directory structure of a well-designed Python package.
- (b) Describe the primary purpose of the argparse module.

module.pg

[] package _ init - .py /

[] package _ atron [nowle.ps /

Readmo, md /

Setip.pj

requirements.fxt

usage_example.rpyhb/

test.py

4. (5 points) Testing Python modules

- (a) Why is it important to add testing classes to Python modules?
- 2 (b) Write set-up and tear-down test classes using pytest for the class below. Your test class should check if the time_of_flight method returns an expected value.

```
import numpy as np
class ParabolicMotion:
    """ A class to calculate the flight time of a projectile."""

def __init__(self, v0, angle):
    """Initial velocity (v0 in m/s) and launch angle (degrees)."""

self.v0 = v0
    self.angle = np.radians(angle)
    self.g = 9.81 # Gravity (m/s^2)

def time_of_flight(self):
    """Returns the total time the projectile stays in the air."""
    return (2 * self.v0 * np.sin(self.angle)) / self.g
```

0) To prevent tre code to break if the user miss over it growing it wrong paremeters or variables / Refactoring code.

Validating code

to-o-f.

it provides to-o-f 20?

raise Value Error as e:

Print ('e'; "Inital velocitions to be a positive homoer") X