Loops & Orbits — Midterm

Friday, Jan. 17, 2020

Part I — Physics & Math

1. Finite Difference Methods

The following is a definition:

$v_{i \to i+1} \equiv \frac{\Delta x}{\Delta t}$

One of these related things is ***not*** true. Circle the letter for the one that is ***not*** true.

A. $\Delta x \equiv x_{i+1} - x_i$
B. $\Delta t \equiv t_{i+1} - t_i$
C. $x_{i+1} = x_i + (t_{i+1} - t_i) \cdot v_{i \to i+1}$
D. $v_{i \to i+1}$ is called the acceleration.
E. For simplicity in notation, sometimes we shorten $v_{i \to i+1}$ to just v_i, and then an equation we have is, $x_{i+1} = x_i + (t_{i+1} - t_i) \cdot v_i$.

2. Constant Acceleration (Drag Racer)

For the drag racer without drag we had the following equation for constant acceleration:

$v_{i+1} = v_i + (t_{i+1} - t_i) \cdot a$

If a is 12 $\frac{\text{m}}{\text{s}^2}$ and $\Delta t = t_{i+1} - t_i$ is 0.5s and $v_3 = 18\ \frac{\text{m}}{\text{s}}$, what is v_4?

A. 21 $\frac{\text{m}}{\text{s}}$
B. 24 $\frac{\text{m}}{\text{s}}$
C. 27 $\frac{\text{m}}{\text{s}^2}$
D. 30 $\frac{\text{m}}{\text{s}}$

3. Baseball with Drag

Air resistance added drag to the baseball. Here is the formula for the strength of the acceleration due to the drag:

$\frac{C_{\text{air}}}{m} s^2$

In this formula, C_{air} is the coefficient of air, m is the mass of the baseball, and s is its speed.

If the baseball is moving with angle θ from horizontal, the baseball has *horizontal* acceleration of:

A. $\frac{C_{\text{air}}}{m} s^2 \sin\theta$

B. $-\frac{C_{\text{air}}}{m} s^2 \sin\theta - g$

C. $\frac{C_{\text{air}}}{m} s^2 \cos\theta$

D. $-\frac{C_{\text{air}}}{m} s^2 \cos\theta$

E. $-\frac{C_{\text{air}}}{m} s^2 \cos\theta - g$

4. Newton

In class, I claimed that one of Newton’s brilliant leaps forward was realizing that _____________ was the important thing to study and express in physical law.

Blank is:

A. friction

B. position

C. velocity

D. acceleration

E. pressure

5. Length from Components

If a vector has components v_x and v_y, then that vector has length

A. $v_x + v_y$

B. $v_x \cdot v_y$

C. $v_x^2 + v_y^2$

D. $\tan^{-1} \frac{v_y}{v_x}$

E. $\sqrt{v_x^2 + v_y^2}$

Part II — Computer Science

6. If Statements

In most programming languages, including Python, if statements are an example of:

A. Expressions

B. Comparisons

C. Control Flow

D. Assignments

E. Library Functions

7. Sequence Types

In Python, we have three ways of creating and passing around sequences:

A. List
B. Range
C. Tuple

Only one of this is mutable, and it can be mutated with append. Choose the sequence type that is mutable.

8. Pass-by-Reference and Pass-by-Value

In Python, functions arguments are pass-by-value. This means that that the function gets a copy of each argument, rather than a reference to the original variable. However, I warned everyone that one type acts as if it is pass-by-reference:

A. Booleans
B. Integers
C. Floats
D. Strings
E. Lists

9. Functions and Scoping

If you create a new variable inside a function definition, that variable is:

A. Available only after the function returns.
B. Available to any caller of the function.
C. Exists and is available only to the code inside the function definition.
D. Can have the same name as a variable outside the function definition in which case we say it “shadows” that variable.
E. C and D.
F. All of A, B, C, and D

10. Software Development Best Practices

I have advocated in class **all but one** of the following as best practices:

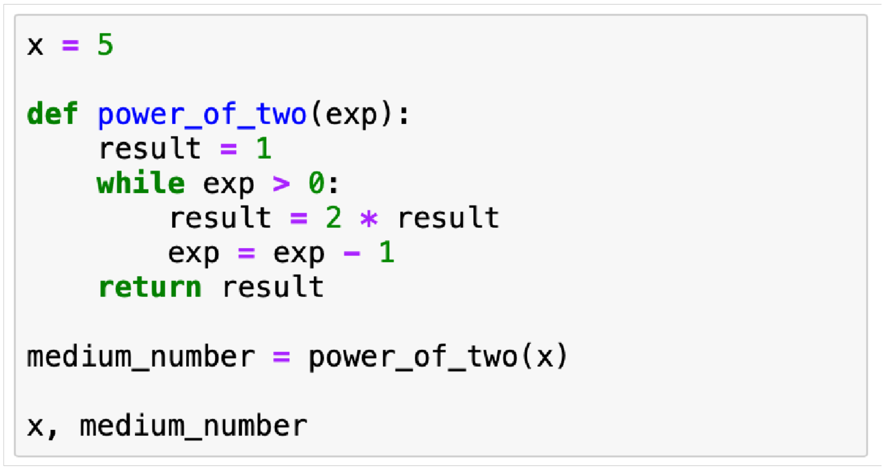
A. Documentation of code (with both Markdown and comments).
B. Use of white space (such as spaces around operators) to make code conventional and easy to read, even though Python does not require it.
C. Using expressive variable and function names so that code is self-documenting.
D. Keeping code very brief and fitting it onto as few lines as possible.
E. Breaking functionality down into small testable parts.
F. Testing each functional part with unit tests.
G. Studying the stack traces and error messages the interpreter spits out.
H. Learning to use the debugger and other professional code development tools.

Part III — Coding

11. Pass-by-Value

What tuple will the following print if it were executed as a cell in a Jupyter notebook?

A. (5, 32)
B. (0, 32)
C. (5, 16)
D. (0, 16)
E. None of the above.

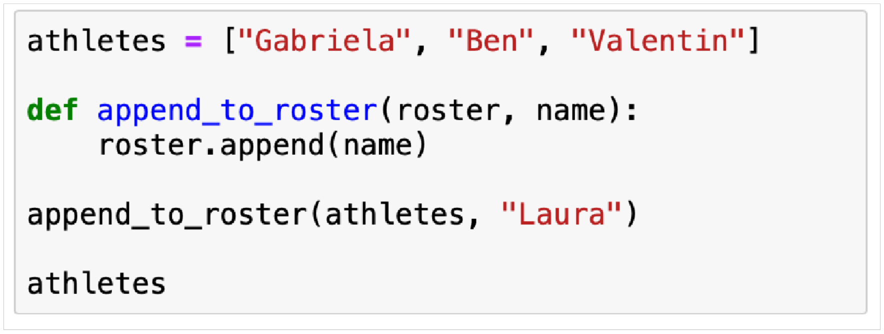


```python
x = 5

def power_of_two(exp):
    result = 1
    while exp > 0:
        result = 2 * result
        exp = exp - 1
    return result

medium_number = power_of_two(x)

x, medium_number
```

12. Pass-by-Reference

What list will the following print if it were executed as a cell in a Jupyter notebook?

A. [“Gabriela”, “Ben”, “Valentin”, “Laura”]
B. [“Laura”, “Gabriela”, “Ben”, “Valentin”]
C. [“Gabriela”, “Ben”, “Valentin”]
D. [“Laura”]
E. None of the above.

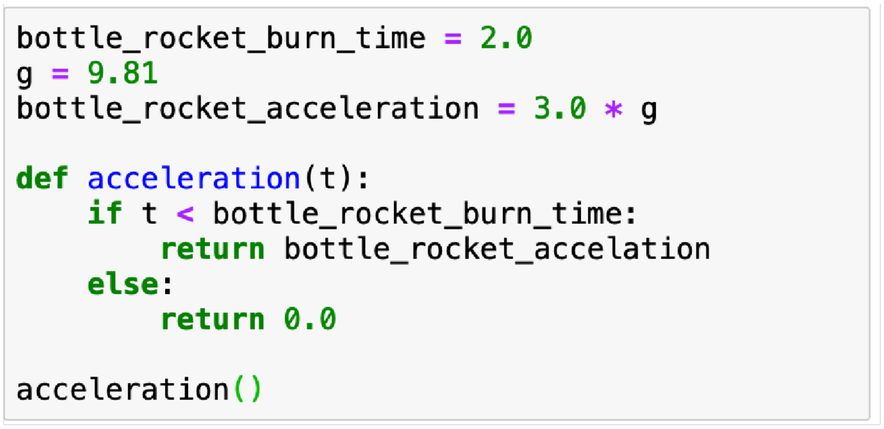


```
athletes = ["Gabriela", "Ben", "Valentin"]

def append_to_roster(roster, name):
    roster.append(name)

append_to_roster(athletes, "Laura")

athletes
```

13. Errors

What error will the following print:

A. bottle_rocket_acceleration is not defined
B. cannot compare two floats
C. function acceleration missing required argument
D. No error.
E. None of the above.



```
bottle_rocket_burn_time = 2.0
g = 9.81
bottle_rocket_acceleration = 3.0 * g

def acceleration(t):
    if t < bottle_rocket_burn_time:
        return bottle_rocket_accelation
    else:
        return 0.0

acceleration()
```

14. Errors

What error will the following print:

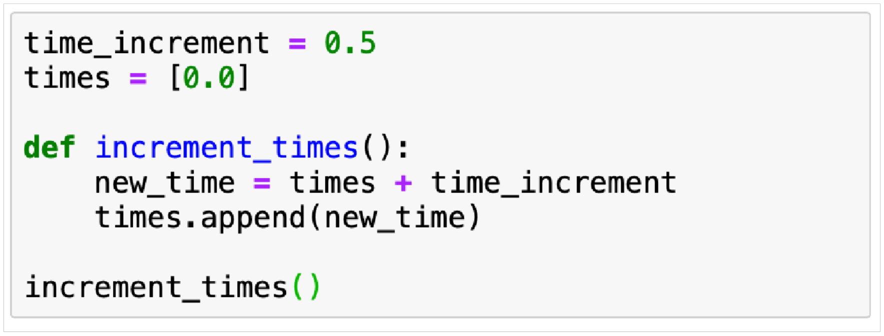
A. time_increment is not defined

B. cannot add float to list

C. function increment_times missing required positional argument

D. No error.

E. None of the above.



```
time_increment = 0.5
times = [0.0]

def increment_times():
    new_time = times + time_increment
    times.append(new_time)

increment_times()
```

15. Ranges

What will the following print:

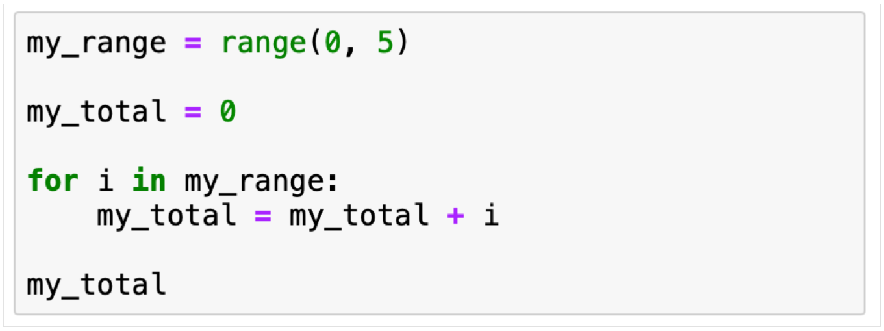
A. 0

B. 5

C. 10

D. 15

E. None of the above.



```
my_range = range(0, 5)

my_total = 0

for i in my_range:
    my_total = my_total + i

my_total
```