

CS116 TUTORIAL 1

INTRODUCTION TO PYTHON

REMINDERS

- Assignment 01 is due Wednesday, January 22nd at 10AM
- Submit Assignment 00 if you have not done so already
- Note: None of your assignments will be marked until you submit A00

REVIEW BASIC PYTHON

- Python is a series of “statements” or commands

```
def f(p1, p2, ..., pn):
```

Statement 1

```
x = 5
```

To make constants: name = value

Statement 2

```
y = 10 * p1 - x
```

```
z = g(x, pn)
```

...

```
...
```

Statement k

```
return z
```

Using `return z` causes the function to return something. All code written after the `return` on the same indentation line will not be executed.

Indentation is extremely important!

REVIEW BASIC PYTHON

Arithmetic operators:

- `+`, `-`, `*`, `/`, `//`, `%`, `**`
- Math module: use `dir(math)` to see list of functions and constants in the module (you have to remember to `import math` first!)

CQ1

What's the result of $(101 // 2) / (5 \% 3)$?

- A. 25.25
- B. 25.0
- C. 25
- D. 20.2
- E. 20

REVIEW PYTHON DESIGN RECIPE

- Don't use: Char or Sym
- New type: Float
- Examples are part of the docstring in the form:

```
'''
```

```
Examples:
```

```
fn_name(p1, p2, ..., pn) => result
```

```
'''
```

- Tests:
 - Make sure to download `check.py` and put the file in the same folder as your assignment questions

```
check.expect("name", fn_name(p1, ..., pn),  
             result)
```

Q1. Write a Python function `create_acct_num` that consumes a positive 3-digit number and returns the corresponding 4-digit number, in which the new last digit is remainder when the sum of the digits of the original number is divided by 7.

- For example,

```
create_acct_num(778) => 7781
```

because $7 + 7 + 8 = 22$ and when 22 is divided by 7, it has remainder of 1

CQ2

- What is the best contract/requirements for `create_acct_num`?

A. `create_acct_num: Num -> Num`

B. `create_acct_num: Int -> Int`
`requires n < 1000`

C. `create_acct_num: Float -> Float`

D. `create_acct_num: Nat -> Nat`
`requires n < 1000`

Q2. Write a Python function `shipping_charges` that calculates cost of shipping boxes.

The function consumes:

- a handling charge (a fixed amount added to each shipment),
- the charge per kg for shipping,
- the weight per box (assumed the same for all boxes), and
- the number of boxes to ship.

The function adds in 13% tax, and returns the total cost.

```
shipping_charges(handling, charge_per_kg,  
                 weight_per_box, num_boxes)
```

For example,

```
shipping_charges(10, 0.25, 10, 5) => 25.425
```

Where 25.425 is computed by performing $(10 + 0.25 * 10 * 5) * 1.13$

Note that the value returned may not be stored exactly. So, your test should not check for an exact value, but rather should check that your answer is very close to the expected value.

Q3. The standard deviation of a set of numbers $x_1, x_2, x_3, \dots, x_n$ is calculated as

$$\sqrt{[(x_1 - x_{mean})^2 + (x_2 - x_{mean})^2 + \dots + (x_n - x_{mean})^2] / n}$$

where x_{mean} is the arithmetic mean (average) of the $x_1, x_2, x_3, \dots, x_n$.

For example, the standard deviation of 3.2, 9.8, 6.3, 1.2, 3.3 is 3.00039997

Write a Python function `std_dev` that consumes five floating point numbers (`x1, x2, x3, x4, x5`) and returns their standard deviation.

Q4. Consider a basic smart phone plan that charges as follows:

- * monthly charge \$39.00
- * \$0.50 per phone minutes after the first 200 free "anytime" minutes
- * \$5.00 per 150 MB of data once the first 100 MB are used.

For example, if you used 15 minutes of phone calls and 87 MB of data, your charge would be \$39.00; if you used 250 minutes of phone calls and 500 MB of data, your charge would be \$79.00 (\$25.0 extra for the phone calls, and \$15.0 extra for data).

Write a Python function `cell_charge` that consumes two natural numbers (`minutes` and `data`) corresponding to the total number of phone minutes used and the amount of data used in a month, and returns the total phone bill for the month, including the basic monthly charge.

Do not perform any rounding on the result.