CS 1112: Introduction To Programming

Supplemental information:

Import in Python

Import

- Python has a TON of optional libraries
  - Libraries are like groups of functions or pre-existing programs you can build from
  - Examples:
    - random - a library for getting pseudo random numbers
    - math - a library with various mathematical functions
    - …
  - We only include the libraries we need

What’s the syntax for importing?

```
import random
```

keyword to say we would like to bring in functions from a module

module/library name

What’s the syntax for importing?

```
import math
```

keyword to say we would like to bring in functions from a module

module/library name

Import Example: importing a library function

Let’s say we want to calculate the number of liters of concrete you need to fill a column

```
import math
# Calculate the volume of a cylinder in liters
def cylinder_volume(radius, height):
    volume = math.pi * radius ** 2 * area * height
    volume = volume * 1000 # Convert from cubic meters into liters
    volume = math.ceil(volume) # Round up
    return volume
v = cylinder_volume(1.5, 10)
```

What’s the syntax?

```
math.ceil(volume)
```

module/library name

the function/constant in the module

What happens if we forget to import?

```python
# Calculate the volume of a cylinder in liters
def cylinder_volume(radius, height):
    volume = math.pi * radius ** 2 * area * height
    volume = volume * 1000 # Convert from cubic meters into liters
    volume = math.ceil(volume) # Round up
    return volume
v = cylinder_volume(1.5, 10)
```

What happens if we forget to import?

NameError: name 'math' is not defined

```
# Calculate the volume of a cylinder in liters
def cylinder_volume(radius, height):
    volume = math.pi * radius ** 2 * area * height
    volume = volume * 1000 # Convert from cubic meters into liters
    volume = math.ceil(volume) # Round up
    return volume
v = cylinder_volume(1.5, 10)
```

Useful imports

math

- **`pi`**
  - `math.pi # 3.141592653589793 (float)`
- **`ceil`** `- rounds up`
  - `math.ceil(3.2) # 4 (int)`
- **`floor`** `- rounds down`
  - `math.floor(3.9) # 3 (int)`
- **`factorial`**
  - `math.factorial(3) # 6 (int) (3 x 2 x 1)`

random

- **`randint`**
  - `random.randint(2, 5) # 2 or 3 or 4 or 5 (int)`

Useful imports

math

A constant!

- **pi**
  - `math.pi # 3.141592653589793 (float)`
- **ceil** - rounds up
  - `math.ceil(3.2) # 4 (int)`
- **floor** - rounds down
  - `math.floor(3.9) # 3 (int)`
- **factorial**
  - `math.factorial(3) # 6 (int) (3 x 2 x 1)`

random

- **randint**
  - `random.randint(2, 5) # 2 or 3 or 4 or 5 (int)`

Question- ceil: What’s the value and type?

```
math.ceil(3.5)
```

Question- ceil: What's the value and type?

```
math.ceil(3.5)
```

4, int

Question- floor: What’s the value and type?

```
math.floor(3.5)
```

Question- floor: What’s the value and type?

```
math.floor(3.5)
```

3, int

Question- factorial: What’s the value and type?

```
math.factorial(4)
```

Question- factorial: What’s the value and type?

```
math.factorial(4)
```

24, int

4 x 3 x 2 x 1 =
4 x 3 x 2 =
4 x 6 =
24