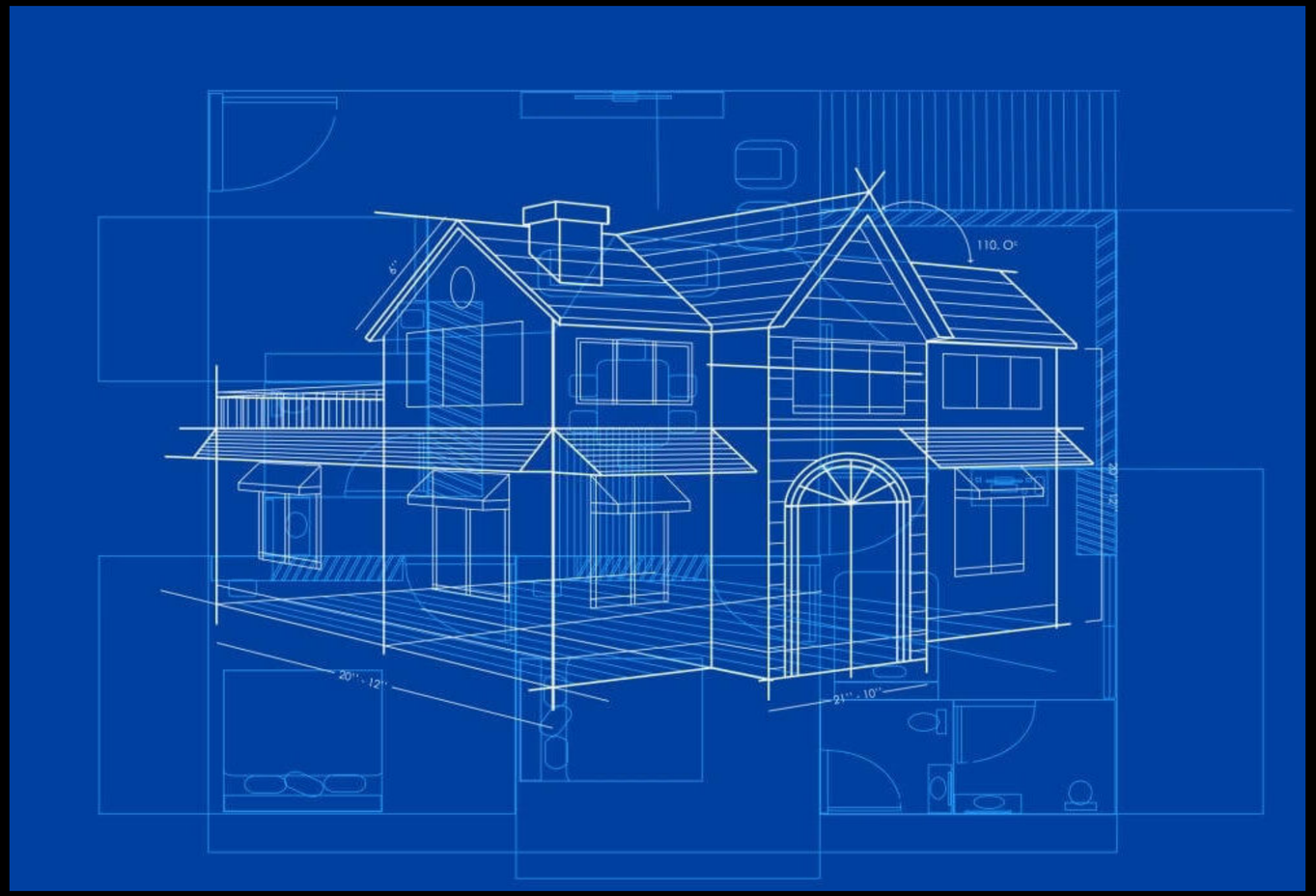
UCSF Intro To Programming

(AKA: Introduction to Computing for Biophysicists / Programming Fundamentals)

Object Oriented Programming (OOP)


110.0°
8'
20'' - 12''
21'' - 10''



Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```
my_home = House(                                                    )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```
my_home = House( home_dimentions, home_window_count, home_door_count )
```

```
home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```
home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```
class House:
```

```
home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```python
class House:
    def __init__(self, dimentions, window_count, door_count):



home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```python
class House:
    def __init__(self, dimentions, window_count, door_count):




    def find_square_footage(self):



    def get_window_count(self):



home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

```python
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):



    def get_window_count(self):



home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

- Dimensions
- Square Footage
- Window Count
- Door Count

...

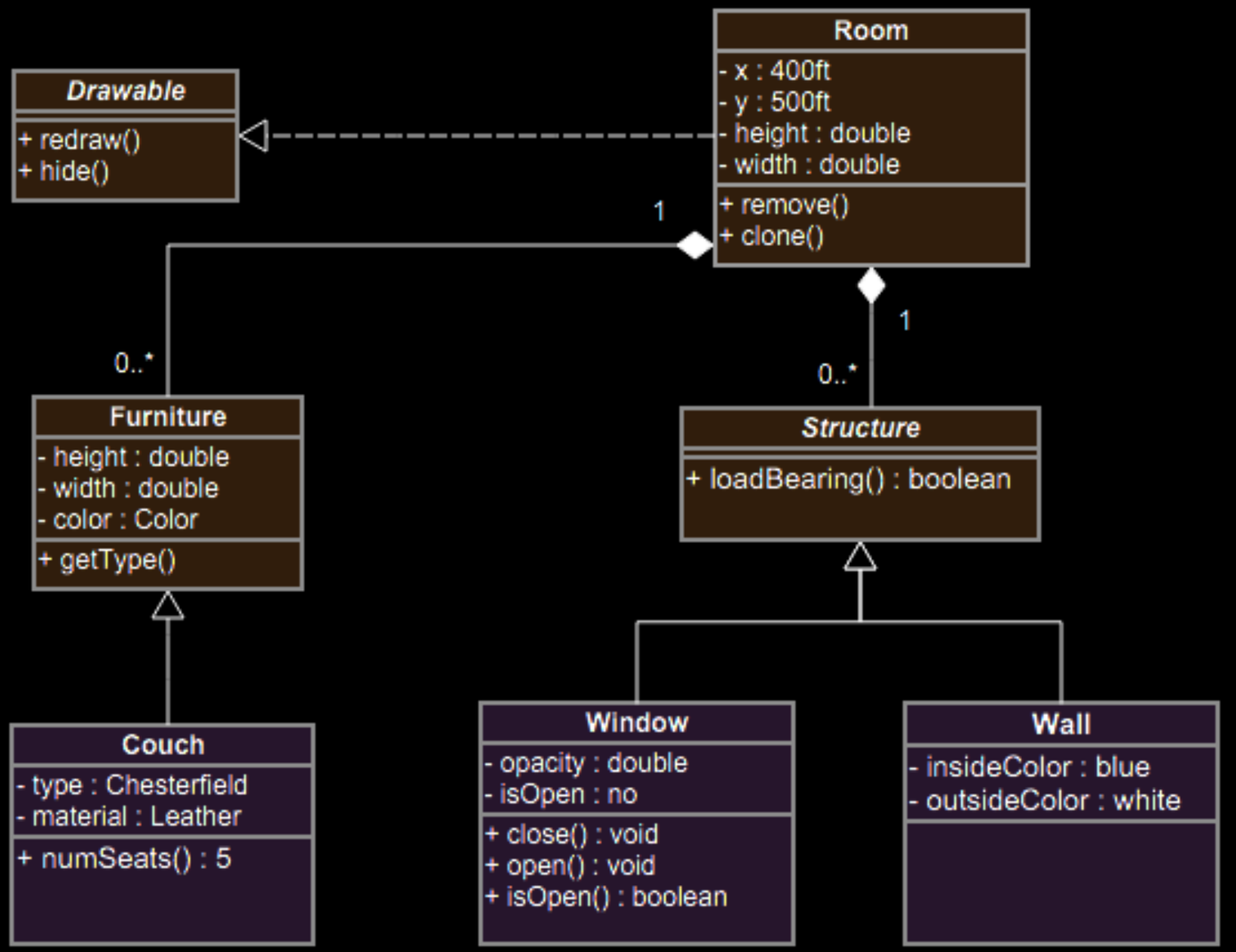
```python
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):
        square_footage = dimentions[0] * dimentions[1]
        self.square_footage = square_footage
        return square_footage

    def get_window_count(self):
        return self.window_count



home_dimentions = [10,10]
home_window_count = 5
home_door_count = 2

my_home = House( home_dimentions, home_window_count, home_door_count )


print( 'My house has this many windows:', my_home.get_window_count() )
```

Attributes to a house:

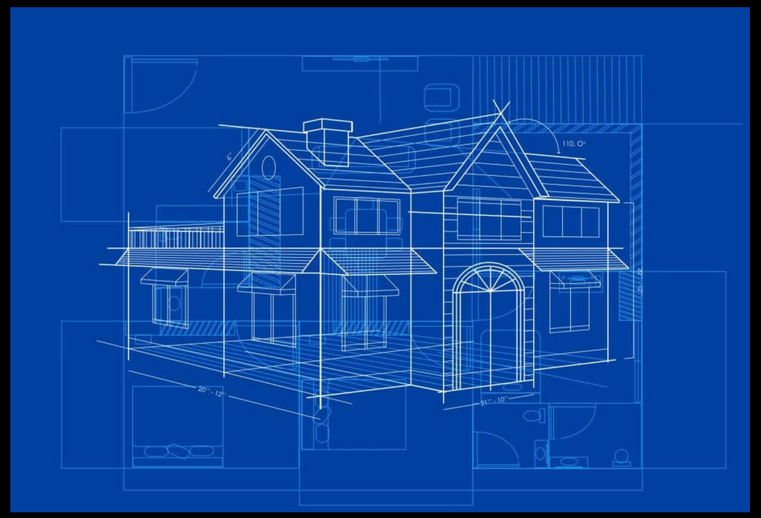
- Dimensions
- Square Footage
- Window Count
- Door Count

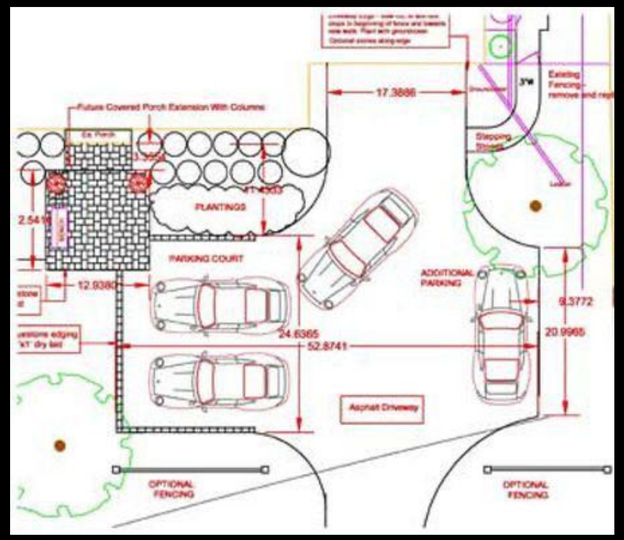
...

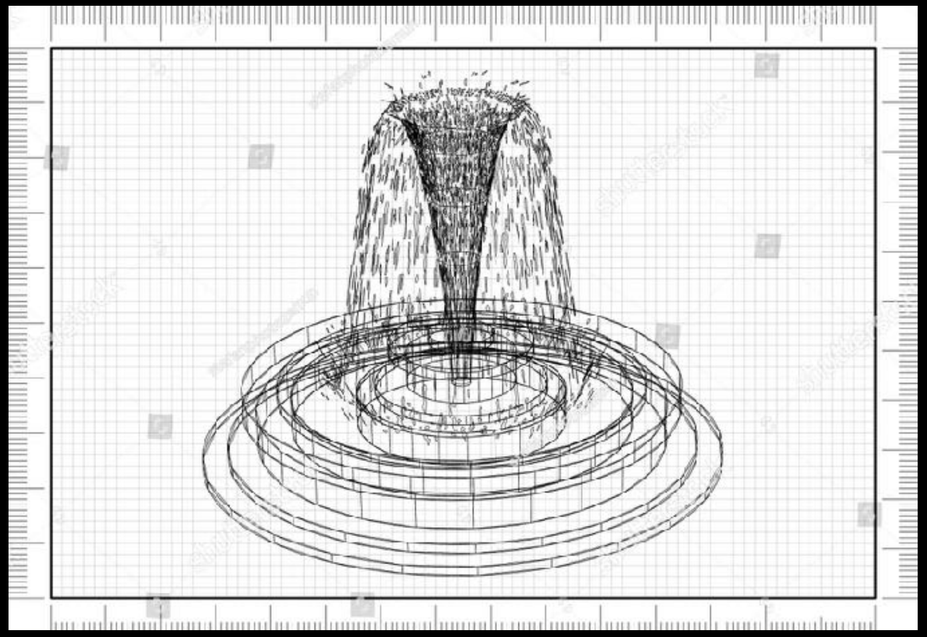

IDEA
IDEA
RIGHT


Drawable
+ redraw()
+ hide()
Room
- x : 400ft
- y : 500ft
- height : double
- width : double
+ remove()
+ clone()
1
0..*
1
0..*
Furniture
- height : double
- width : double
- color : Color
+ getType()
Structure
+ loadBearing() : boolean
Couch
- type : Chesterfield
- material : Leather
+ numSeats() : 5
Window
- opacity : double
- isOpen : no
+ close() : void
+ open() : void
+ isOpen() : boolean
Wall
- insideColor : blue
- outsideColor : white

Packages

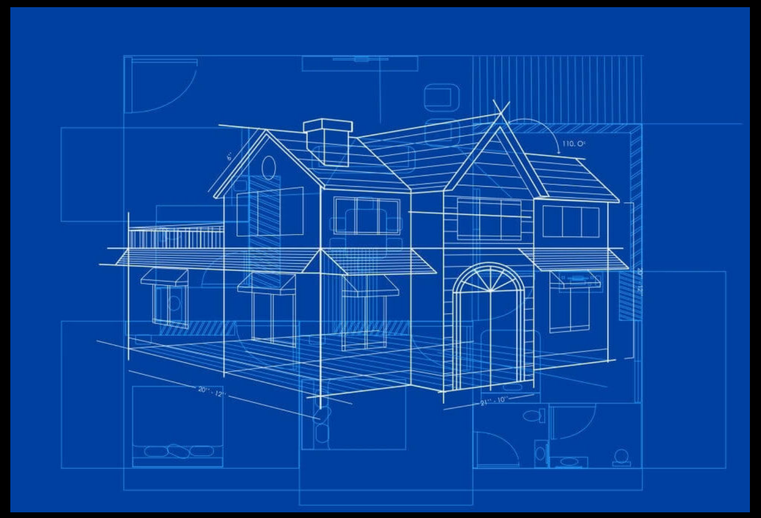


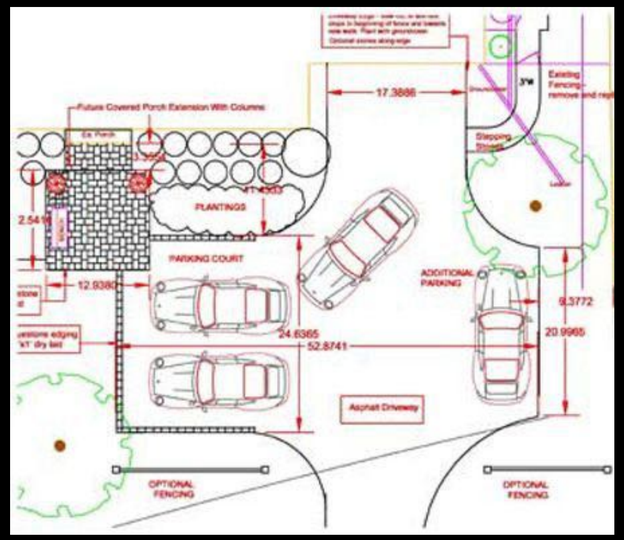

Future Covered Porch Extension With Columns
17.3888
PLANTINGS
PARKING COURT
12.9380
24.6365
52.8741
ADDITIONAL PARKING
9.3772
20.9965
Asphalt Driveway
OPTIONAL FENCING
OPTIONAL FENCING
Existing Fencing - remove and
Stepping Stones

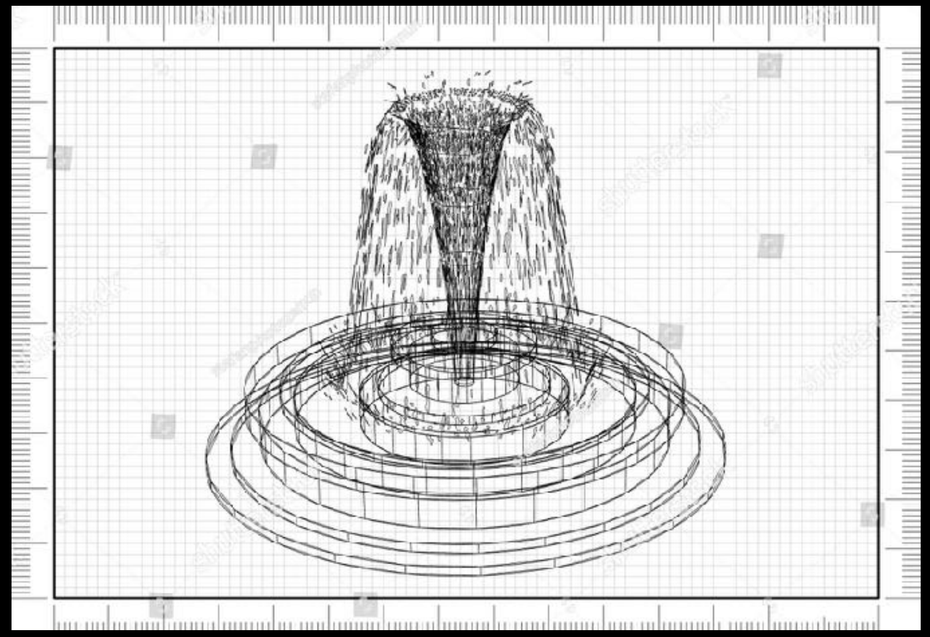


Civil Engineer Package

civil_engineering







buildings.py

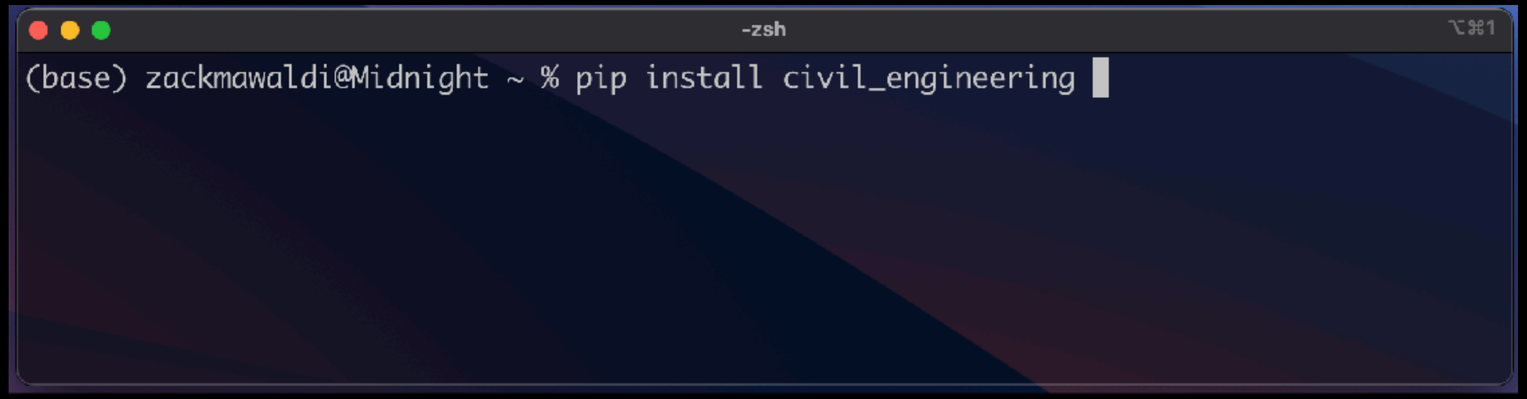
```
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):
        square_footage = dimentions[0] * dimentions[1]
        self.square_footage = square_footage
        return square_footage

    def get_window_count(self):
        return self.window_count
```

buildings.py

```
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):
        square_footage = dimentions[0] * dimentions[1]
        self.square_footage = square_footage
        return square_footage

    def get_window_count(self):
        return self.window_count
```

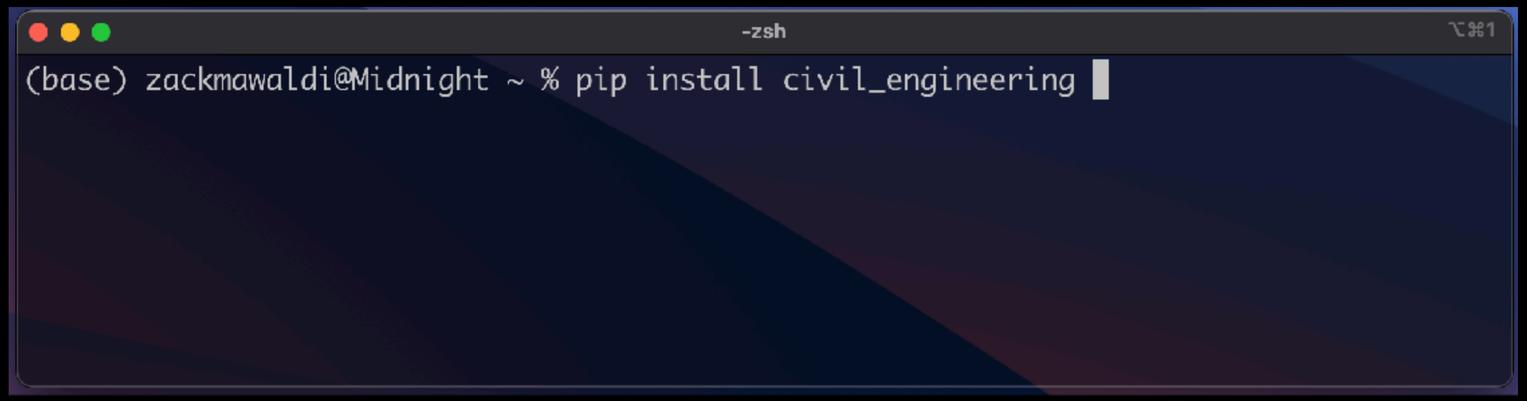
buildings.py

```python
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):
        square_footage = dimentions[0] * dimentions[1]
        self.square_footage = square_footage
        return square_footage

    def get_window_count(self):
        return self.window_count
```



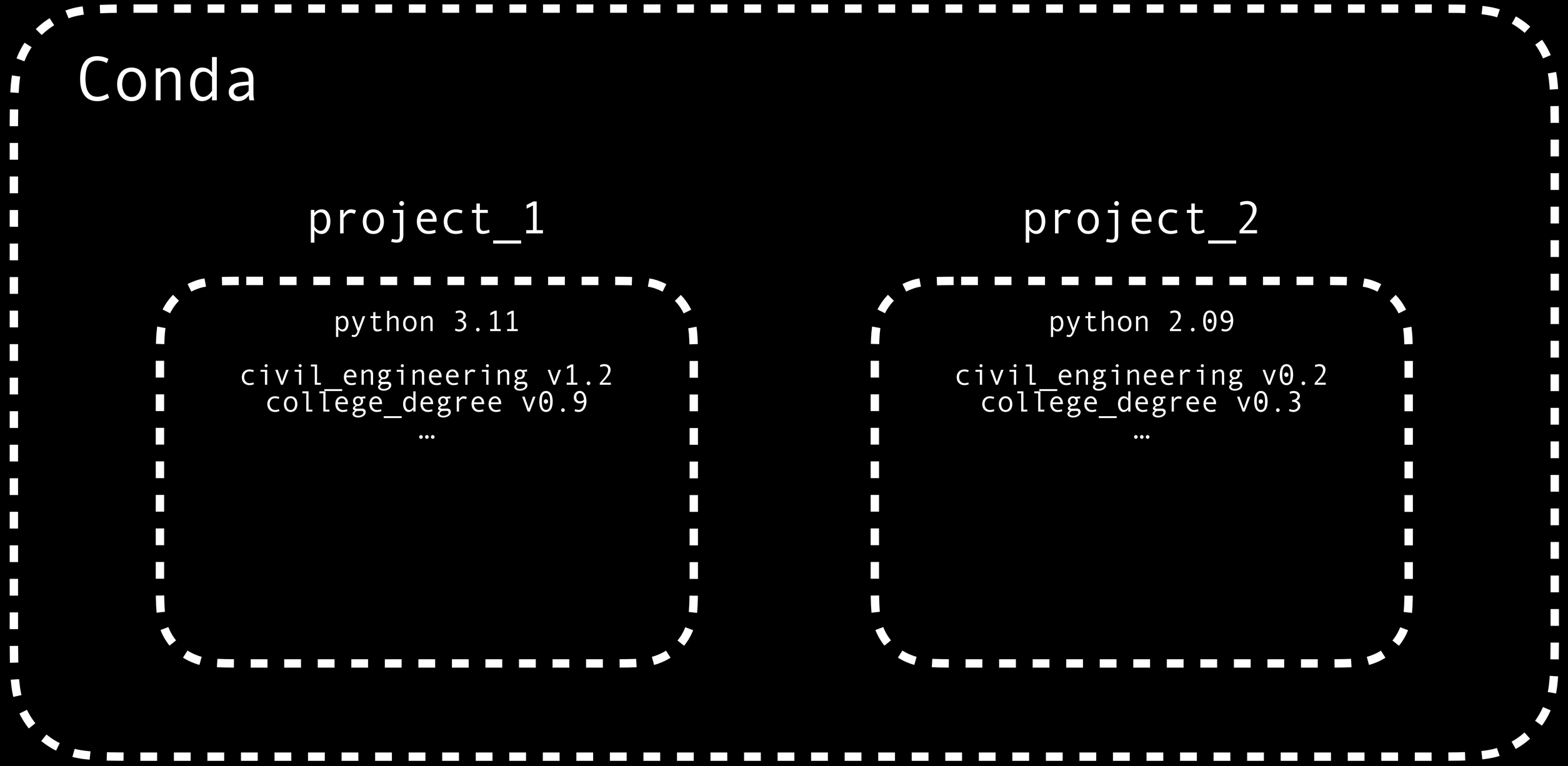
buildings.py

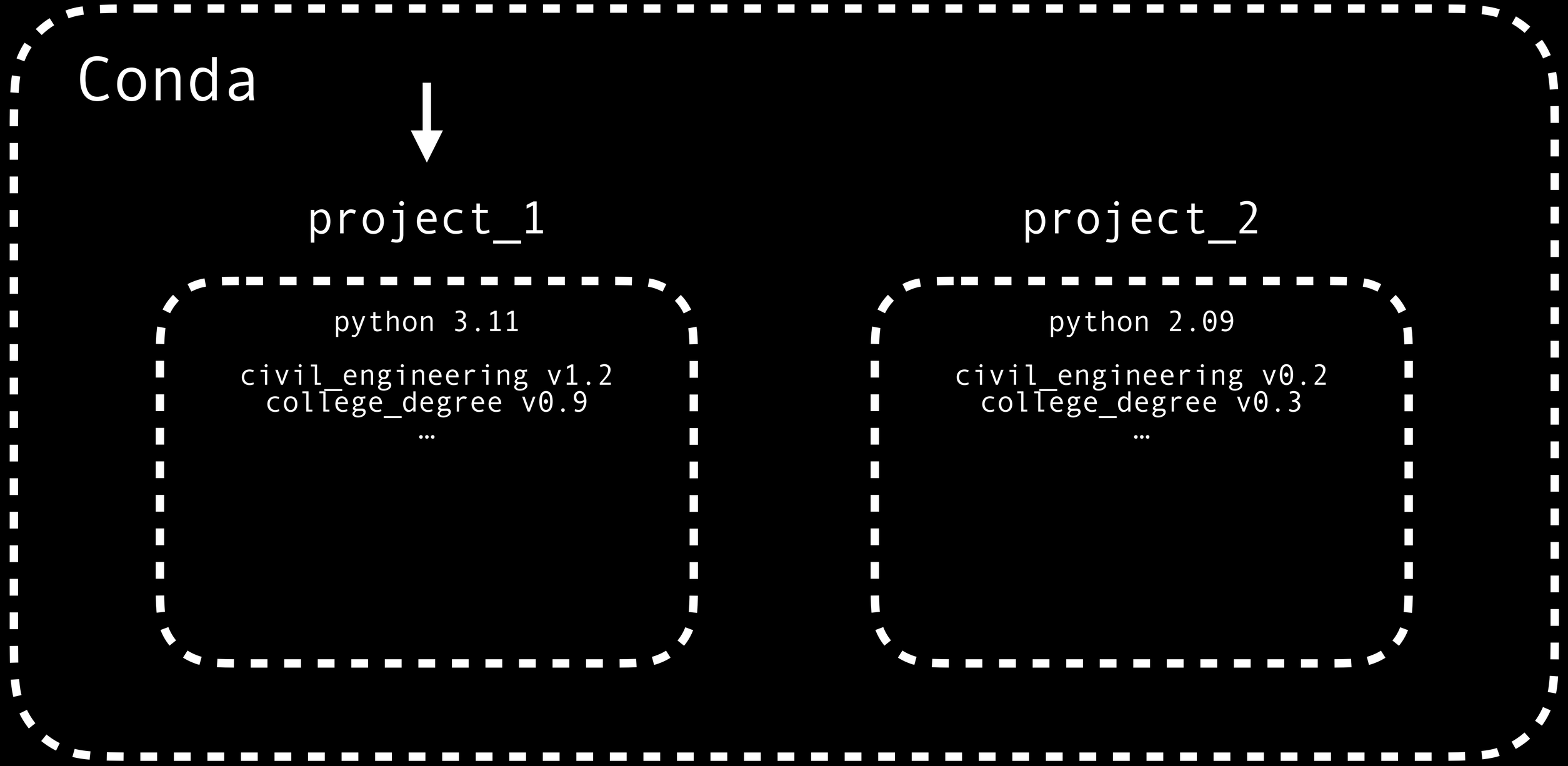
```python
class House:
    def __init__(self, dimentions, window_count, door_count):
        self.dimentions = dimentions
        self.window_count = window_count
        self.door_count = door_count
        self.square_footage = -1

    def find_square_footage(self):
        square_footage = dimentions[0] * dimentions[1]
        self.square_footage = square_footage
        return square_footage

    def get_window_count(self):
        return self.window_count
```




```python
from civil_engineering.buildings import House

my_home = House( home_dimentions, home_window_count, home_door_count )
```

Conda
(Package Manager)


Conda
project_1
python 3.11
civil_engineering v1.2
college_degree v0.9
…
project_2
python 2.09
civil_engineering v0.2
college_degree v0.3
…


Conda
project_1
python 3.11
civil_engineering v1.2
college_degree v0.9
…
project_2
python 2.09
civil_engineering v0.2
college_degree v0.3
…