week 5

notes

Variable: protected and private

- Protected

By prefixing with single underscore.Does *not* change anything, just a convention!

- **Private**

By prefixing with at least two underscores and suffixed with at most one underscore

setters and getters

- “getters” and “setters” methods are often use to modify the class variable/property.
- as opposed to referring it by:

InstanceName.Instance.Variable

- the getters use “return”. previous simple method examples “print”. this way the resulting value cannot be assigned to a variable. with return you can assign the result to a variable

examples:

```
def setname(self, fullname):
    self.name = fullname

def getname(self):
    return self.name
```

static method

using static methods, declared by using @staticmethod “decorators”, you can call a method by its class Name, without even instantiating an object. There are cases where you want to do this.

(it’s a bit like “helper class in Java)

examples:

```
@staticmethod
def displayCountess():
    return Employee.empCount
```

(if you use this in “Employee” class)

you can then call this method by:

Employee.displayCountess()

__init__

remember __init__ method is a special one, it gets executed every time you instantiate an object. you could use this to keep track something/count as objects are get created

again, dictionary!!

remember you can define an empty dictionary

```
a={}
```

and keep adding key/values by specifying

```
a[key]=value
```

Importing

There are (at least) 2 ways to import external file (module), for example you have a python file <test.py> that has a class called check

1. import test
2. from test import check

Importing(cont)

1. #1 you need to call the class with module name prefix : test.class
2. #2 you can directly call class