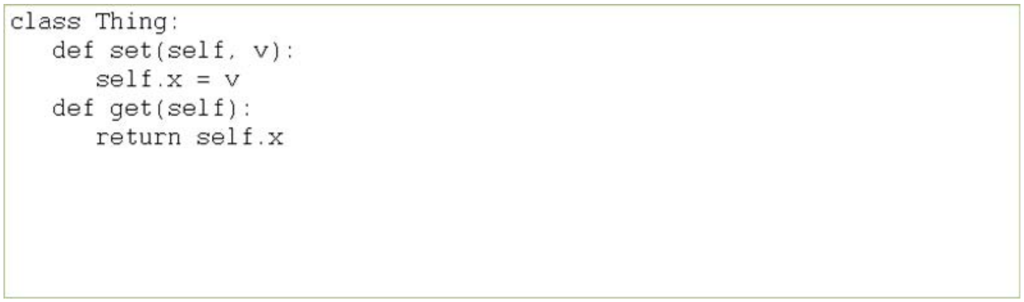
Problem Wk.1.4.6: OOPs

Part 1: Thing clone

Add a method called `clone` to the `Thing` class which returns a completely new `Thing` with the same stored value.

```
>>> a = Thing()
>>> a.set(3)
>>> b = a.clone()
>>> b.get()
3
```

Use the `set` and `get` methods of `Thing`, do not access `x` directly.



```
class Thing:
   def set(self, v):
      self.x = v
   def get(self):
      return self.x
```

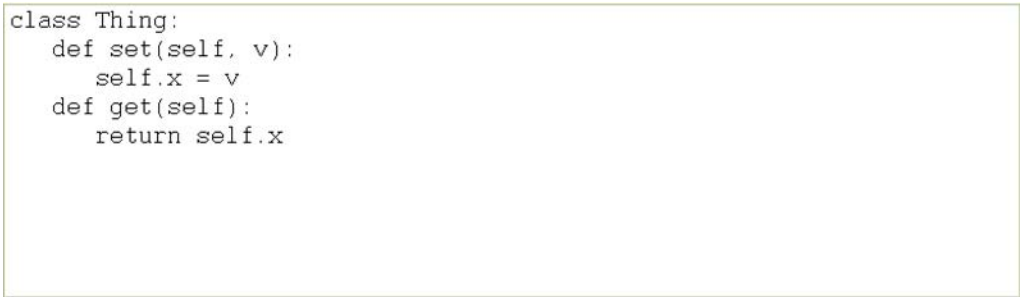
Part 2: Thing str

Add a `__str__` method to the `Thing` class so that printing a `Thing` instance generates an informative string.

```
>>> a = Thing()
>>> a.set(3)
>>> print a
This is a Thing with value 3
```

Use the `set` and `get` methods of `Thing`, do not access `x` directly.

Python has a built-in function `str` that will be useful. Look up its definition in the documentation.



```
class Thing:
   def set(self, v):
      self.x = v
   def get(self):
      return self.x
```

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011