

# OOP - Lesson 2

Instances of Objects

# What is an Instance of an Object?

- Create an instance any time you want to use the object
- For example, we coded a WashingMachine. Now I want to use a WashingMachine. I do that by creating an instance of the WashingMachine class. This is called **instantiation**.
- To further illustrate, if I have 2 washing machines in my house, I would code 1 class WashingMachine, and I would create 2 different instances of the WashingMachine class.

# Syntax for creating an instance

- To instantiate a WashingMachine object, do this:

```
WashingMachine(60, 700, "cottons", False, "persil")
```

- Remember to pass in any parameters that you defined in your `__init__()` method of the WashingMachine class
- If you want to be able to use your WashingMachine object, set it to a variable:

```
myWashingMachine = WashingMachine(60, 700, "cottons", False, "persil")
```

## Syntax for creating an instance (cont)

- If you have 2 washing machines, instantiate 2 WashingMachine objects

- 

```
my_american_washing_machine = WashingMachine(60, 700, "cottons", False, "persil")  
my_israeli_washing_machine = WashingMachine(60, 900, "delicates", False, "ariel")
```

# Class vs Instance Attributes


- The attributes that we defined so far on our washing machine (temp, spin\_speed etc) are called Instance Attributes.
- That means that each time you instantiate a new WashingMachine object, those values will be different.
- What if I want an attribute that all WashingMachines share? For example, a prerequisite for having a washing machine is that it can connect to a water pipe. This is different from another machine - like a refrigerator - which doesn't need a connection to a water pipe.

# Class Attribute Syntax

```
class WashingMachine:  
    prerequisite = "water_pipe"  
  
    #constructor  
    def __init__(self, temp, spin_speed, cycle):  
        self.temp = temp  
        self.speed = spin_speed
```



Class attribute



Instance attributes

# Accessing Class and Instance Attributes: Dot Notation

They are accessed the same way, via **dot notation**

```
my_washing_machine.prerequisite  
my_washing_machine.temp
```

You can also invoke a method on an instance of an object using the dot notation

```
myWashingMachine.wash_laundry()
```

# Assignment

Write a class that makes Shabbos for you, named ShabbosMaker.

It should have at least 1 class variable, at least 1 instance variable, and at least 3 methods.

Create an instance of your ShabbosMaker and use it! (i.e. cause it to make Shabbos by calling your functions)

Don't forget that as a best practice, your outermost layer of code should be in a function called main()