

COMP  
110

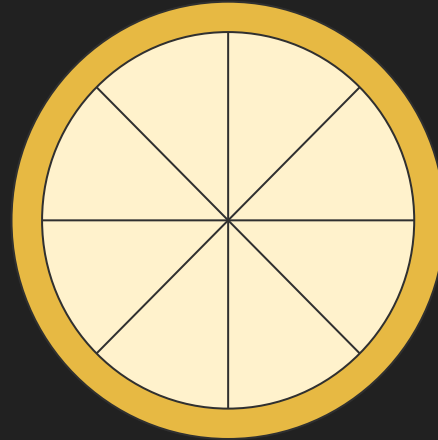
# Object Oriented Programming

# Example: Pizza

size: small

toppings: 0

gluten free: no

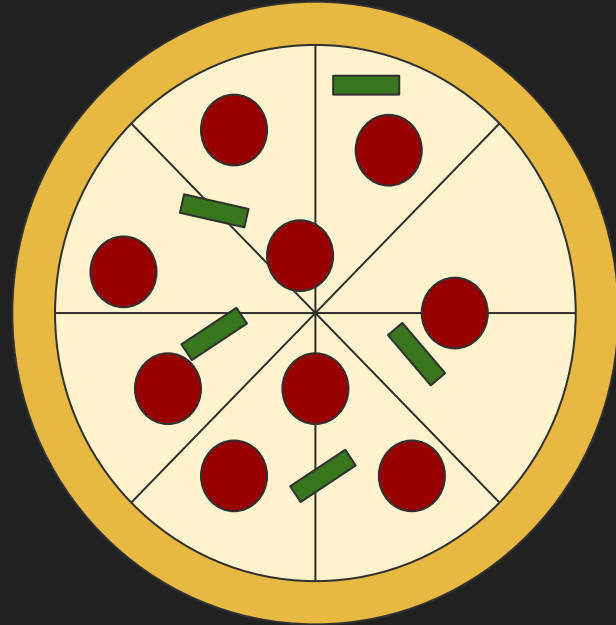


# Example: Pizza

size: large

toppings: 2

gluten free: yes



# Object Oriented Programming

Lets you create new objects in your program.

“Type” ~> “Class”

“Data/Variables” ~> “Attributes”

“Functions” ~> “Methods”

# Attributes

- variables that belong to each instantiation of the object
- Syntax:

<attribute name> : <type>

gluten\_free : bool

# Attributes

- variables that belong to each instantiation of the object
- Syntax:

<attribute name> : <type>

gluten\_free : bool

# Constructor

- Method that defines what happens when new object is created
- Signature Syntax:

```
def __init__(self, <other parameters>):
```

\*Essentially returns **self**

- Instantiation:

```
<class name>(<arguments>)
```



# Try it yourself!

- In `pizza_orders.py`, write a function called `price`
- It should take a `Pizza` object as input and return a `float`
- Basic behavior:
  - If a pizza size is “large”, it should cost \$6.25, otherwise, it should cost \$5
  - Each topping costs \$.75
  - If a pizza is gluten free, add \$1 to the cost.
- Now test the function on `my_pizza`
- Create a new pizza of size “medium”, with 5 toppings, that’s not gluten free and test the function again



# Methods

- Functions that belong to an object
- Calling a method:

```
price(my_pizza) ~> my_pizza.price()
```

- Defining a method:

```
def <method_name>(self, <other parameters>) -> <return type>:
```

```
def price(self) -> float:
```

# Challenge Question

Instructions on website!