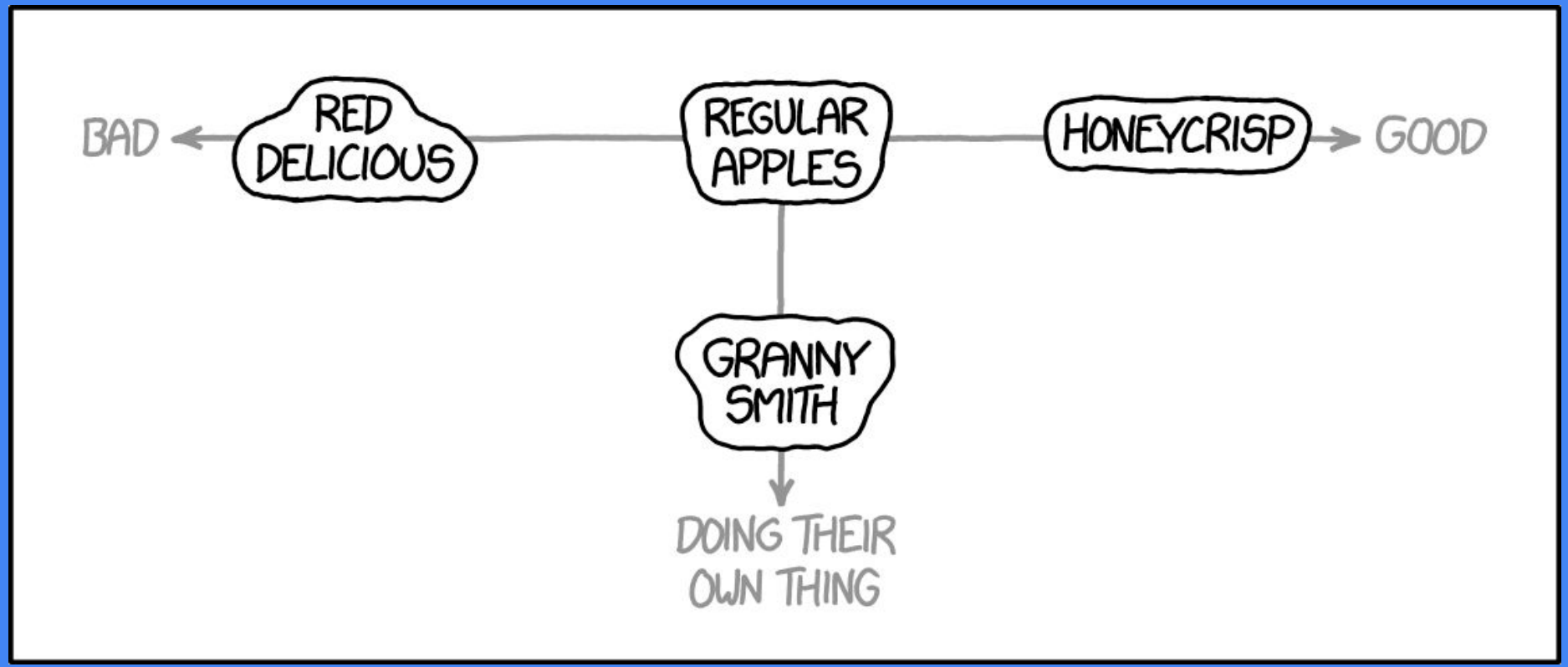
# ITSE-1402 Intermediate Python

## Class 5: Inheritance

June 15, 2017

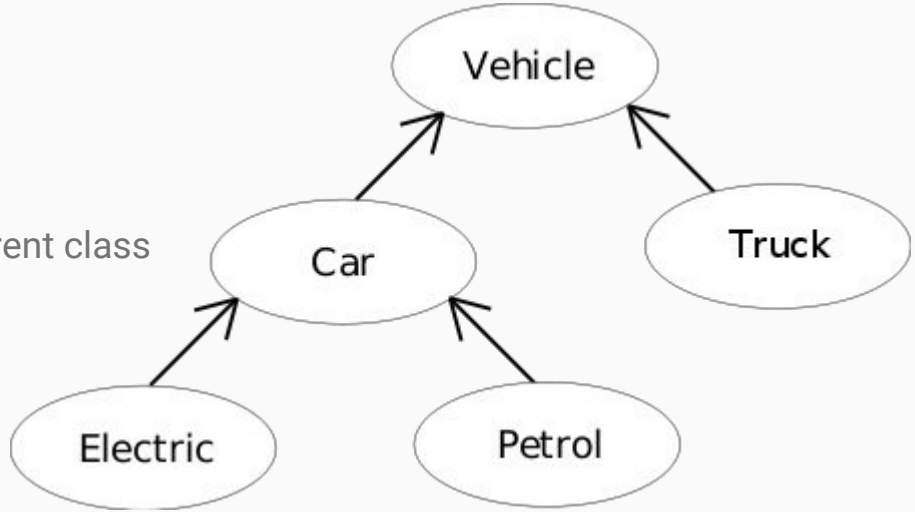



If I were trapped on a desert island, and could have an unlimited supply of any one type of apple, I'd be like, "How did this situation happen?"

https://xkcd.com/1766/

# Hierarchy

- parent class (superclass)
- child class (subclass)
  - Inherits all data and behaviors of parent class
  - Add additional info
  - Add additional behavior
  - Override behavior



# Defining With Inheritance

```
class Vehicle(object):
    def __init__(self, year, make, model):
        self.year = year
        self.make = make
        self.model = model
    def __str__(self):
        return “This is a {} {} {}.”.format(year, make, model)
```

# Defining With Inheritance

```
class Vehicle(object):
        def __init__(self, year, make, model):
                self.year = year
                self.make = make
                self.model = model
        def __str__(self):
                return “This is a {} {} {}.”.format(year, make, model)

class  Car(Vehicle):
        def __str__(self):
                 return “This car is a {} {} {}.”.format(year, make, model)
       def go(self):
                print(“vroom vroom. I’m a car.”)
```

# Defining With Inheritance

```
class Vehicle(object):
        def __init__(self, year, make, model, mileage=None):
                self.year = year
                self.make = make
                self.model = model
        def __str__(self):
                return “This is a {} {} {}.”.format(year, make, model)

class  Car(Vehicle):
        def __init__(self, year, make, model, mileage, body_type):
                Vehicle.__init__(self, year, make, model, mileage)
                self.body_type = body_type
        def __str__(self):
                 return “This car is a {} {} {}.”.format(year, make, model)
       def go(self):
                print(“vroom vroom. I’m a car.”)
```

# Multiple Inheritance

# Multiple Inheritance

```
class Clock(object):
        def __init__(self, hour, minute, second):
                self.hour = hour
                self.minute = minute
                self.second = second

class Calendar(object):
        def __init__(self, day, month, year):
                self.day = day
                self.month = month
                self.year = year

class CalendarClock(Clock, Calendar):
        def __init__(self, day, month, year, hour, minute, second):
                Clock.__init__(self, hour, minute, second)
                Calendar.__init__(self, day, month, year)
```