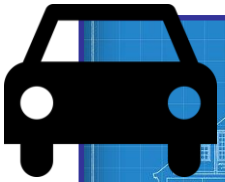
Lecture

Inheritance

Add attributes to the subclass

Inheritance



Vehicle

Class

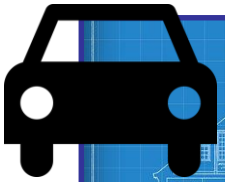
Airplane

Class

Truck

Class

Inheritance



Vehicle

Class

Airplane

Pilot

Class

Truck

Driver

Class

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver


>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver


>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver


>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver
```

```
>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver
```

```
>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, driver):
        self.driver = driver


>>> truck = Truck("M. Johnson")
>>> truck.speed
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    truck.speed
AttributeError: 'Truck' object has no attribute 'speed'
```

Not Automatic

Not Automatic

When the subclass has an __init__() method

Inheritance – Attributes

```
class Superclass:
    # Body

class Subclass(Superclass):

    def __init__(self, <superclass_params>, <subclass_params>):
        Superclass.__init__(self, <superclass_params>)
        self.<attr1> = <value1>
        self.<attr2> = <value2>
        # as many as needed
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
Superclass.__init__(self, <superclass_params>)
```

Inheritance – Attributes

```
class Superclass:
    # Body

class Subclass(Superclass):

    def __init__(self, <superclass_params>, <subclass_params>):
        Superclass.__init__(self, <superclass_params>)
        self.<attr1> = <value1>
        self.<attr2> = <value2>
        # as many as needed
```

Inheritance – Attributes

```
class Superclass:
    # Body

class Subclass(Superclass):

    def __init__(self, <superclass_params>, <subclass_params>):
        Superclass.__init__(self, <superclass_params>)
        self.<attr1> = <value1>
        self.<attr2> = <value2>
        # as many as needed
```

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Include all the attributes from the superclass

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Inheritance – Attributes

```
class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price

class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

Inheritance – Attributes

```
>>> class Vehicle:

    def __init__(self, speed, miles, price):
        self.speed = speed
        self.miles = miles
        self.price = price


>>> class Truck(Vehicle):

    def __init__(self, speed, miles, price, driver):
        Vehicle.__init__(self, speed, miles, price)
        self.driver = driver
```

```
>>> truck = Truck(70, 20000, 25000, "M. Johnson")
>>> truck.speed
70
```

Inheritance – Attributes

```
class Superclass:
    # Body

class Subclass(Superclass):

    def __init__(self, <superclass_params>, <subclass_params>):
        Superclass.__init__(self, <superclass_params>)
        self.<attr1> = <value1>
        self.<attr2> = <value2>
        # as many as needed
```

Inheritance – Attributes

```
class Superclass:
    # Body

class Subclass(Superclass):

    def __init__(self, <superclass_params>, <subclass_params>):
        Superclass.__init__(self, <superclass_params>)
        self.<attr1> = <value1>
        self.<attr2> = <value2>
        # as many as needed
```

Now… Examples