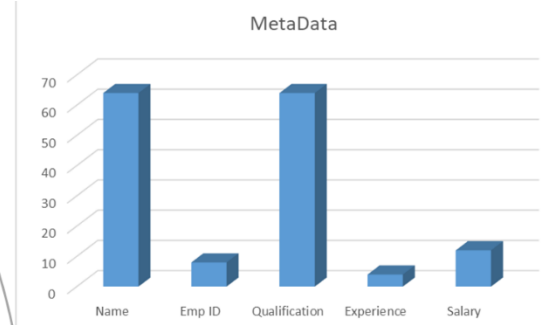
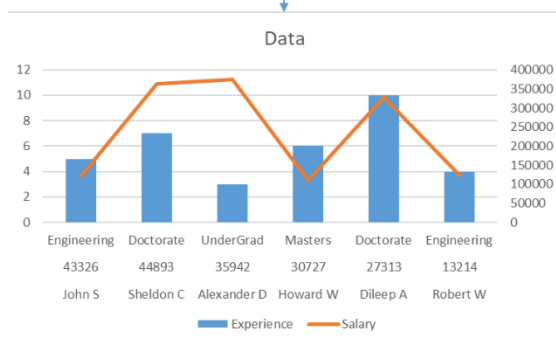


Chapter 1: The Need For and Applications of Meta programming

Name	Emp ID	Qualification	Experience	Salary
John S	43326	Engineering	5	121163
Sheldon C	44893	Doctorate	7	362531
Alexander D	35942	UnderGrad	3	375620
Howard W	30727	Masters	6	110979
Dileep A	27313	Doctorate	10	328056
Robert W	13214	Engineering	4	122947

Name	Emp ID	Qualification	Experience	Salary
John S	43326	Engineering	5	121163
Sheldon C	44893	Doctorate	7	362531
Alexander D	35942	UnderGrad	3	375620
Howard W	30727	Masters	6	110979
Dileep A	27313	Doctorate	10	328056
Robert W	13214	Engineering	4	122947



Representation of metadata

Column	DataType	Size	DataFrame Name
Name	String	64	EmployeeData
Emp ID	Int64	8	EmployeeData
Qualification	String	64	EmployeeData
Experience	Int32	4	EmployeeData
Salary	Float96	12	EmployeeData

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-11-f8ebbea09bb2> in <module>
----> 1 add('meta',1)

<ipython-input-8-1228376a5862> in add(a, b)
      1 def add(a,b):
----> 2     c = a + b
      3     return c

TypeError: can only concatenate str (not "int") to str

```

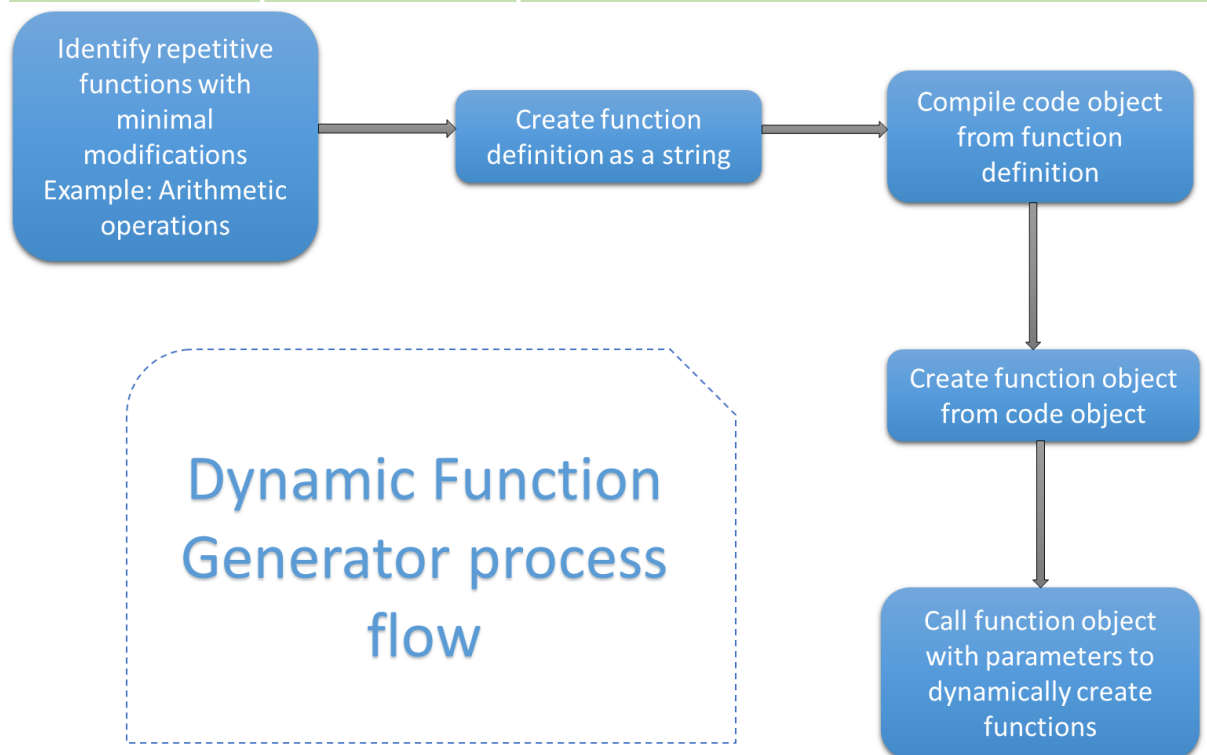
```
def add(a,b):  
    c = a + b  
    return c
```

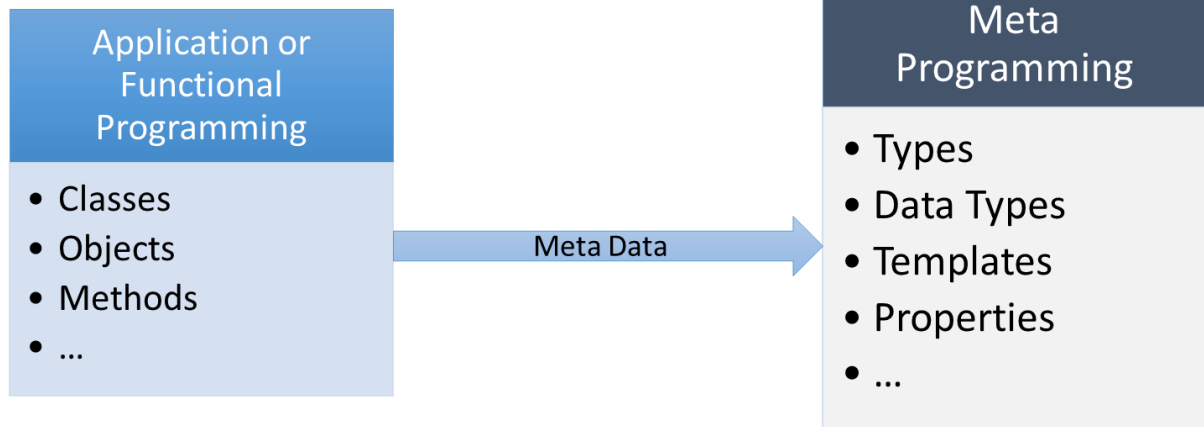
Data	Metadata
add	function
a	Integer or string
b	Integer or string
c	Integer or string

Input a	Input b	Output
integer	integer	integer
string	string	string
integer	string	Message to input both the values as integers or string
string	integer	Message to input both the values as integers or string

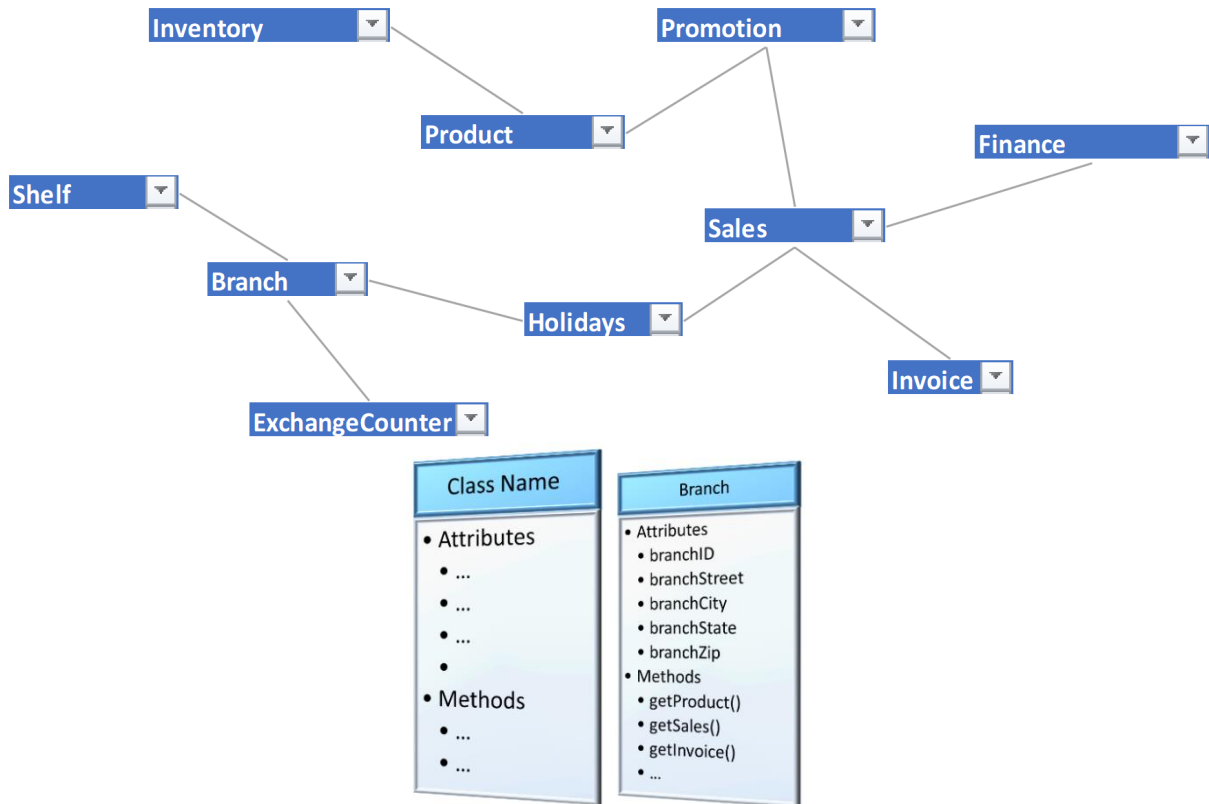
Input a	Input b	Output
integer	integer	integer
string	string	string
integer	string	string
string	integer	string

Input a ▼	Input b ▼	Output ▼
integer	integer	integer
float	float	float
integer	float	float
float	integer	float
string	string	Message to input both values as numbers
integer	string	Message to input both values as numbers
string	integer	Message to input both values as numbers



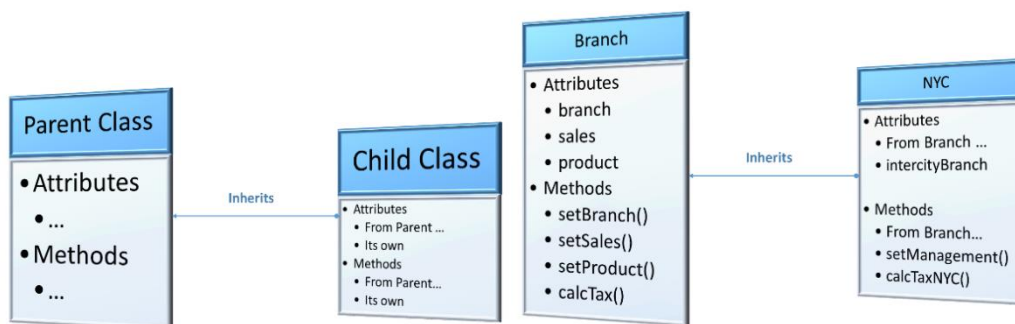


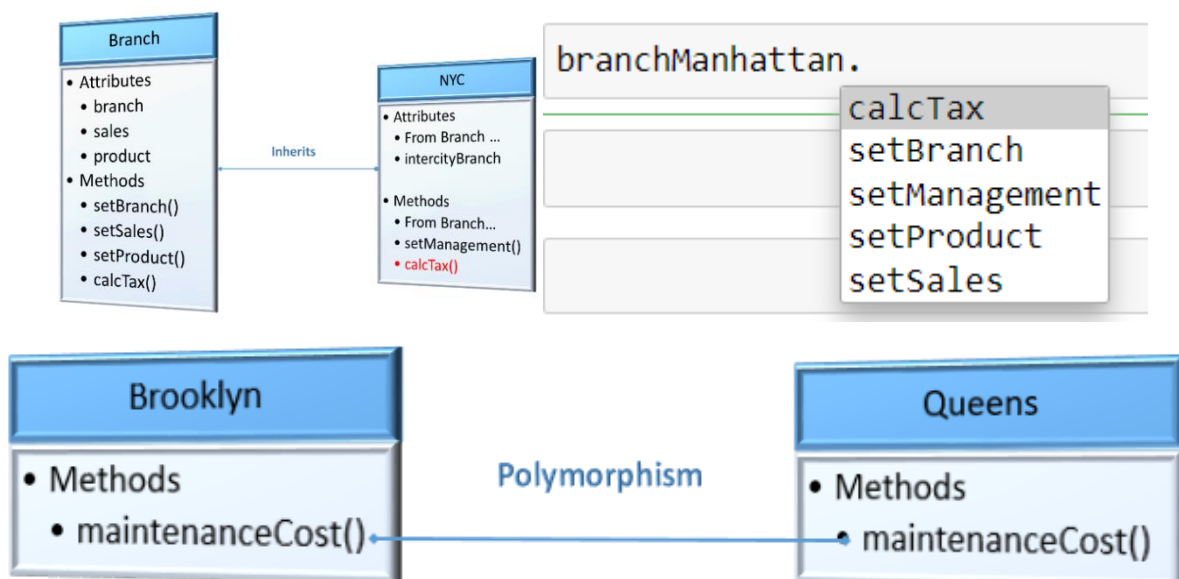
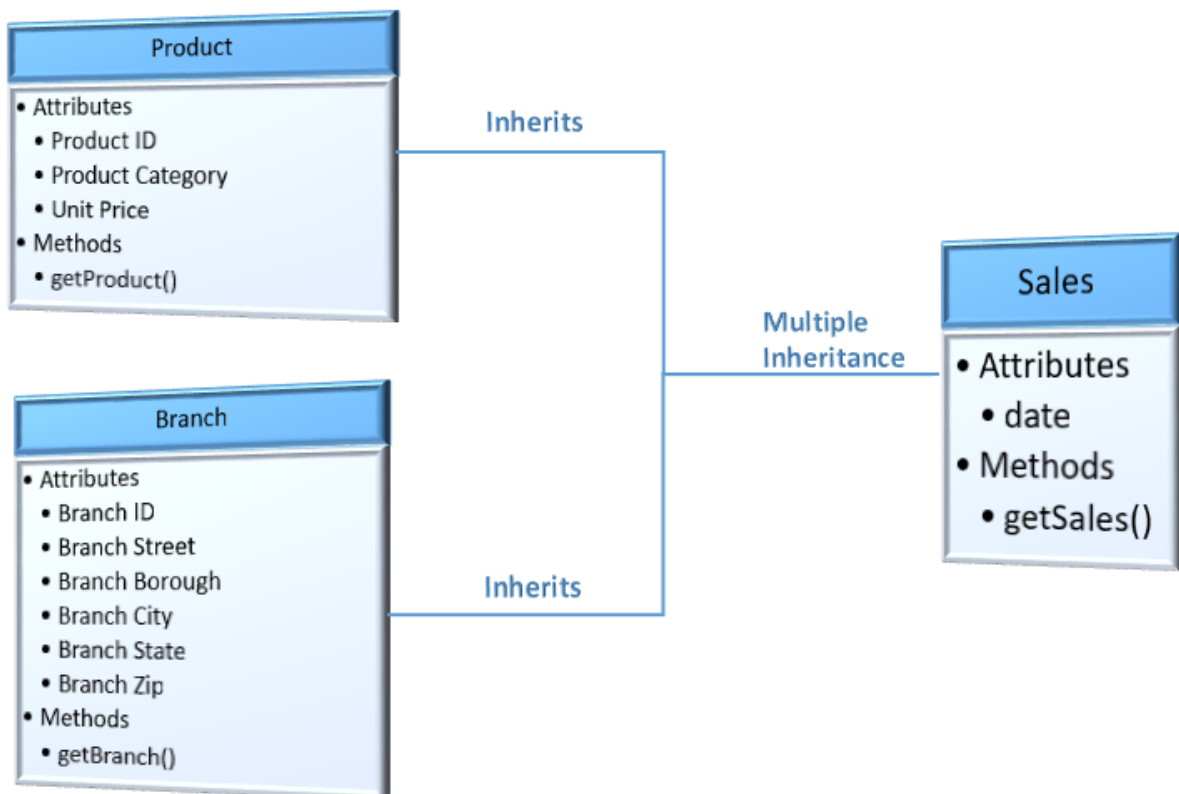
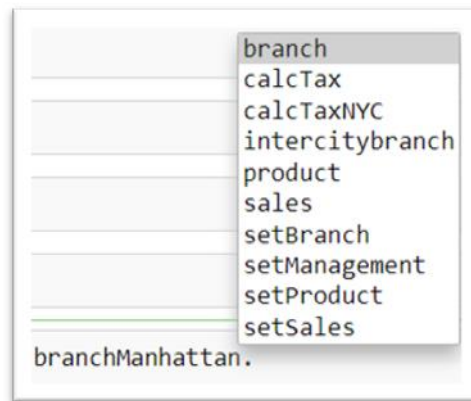
Chapter 2: Refresher of OOP Concepts in Python

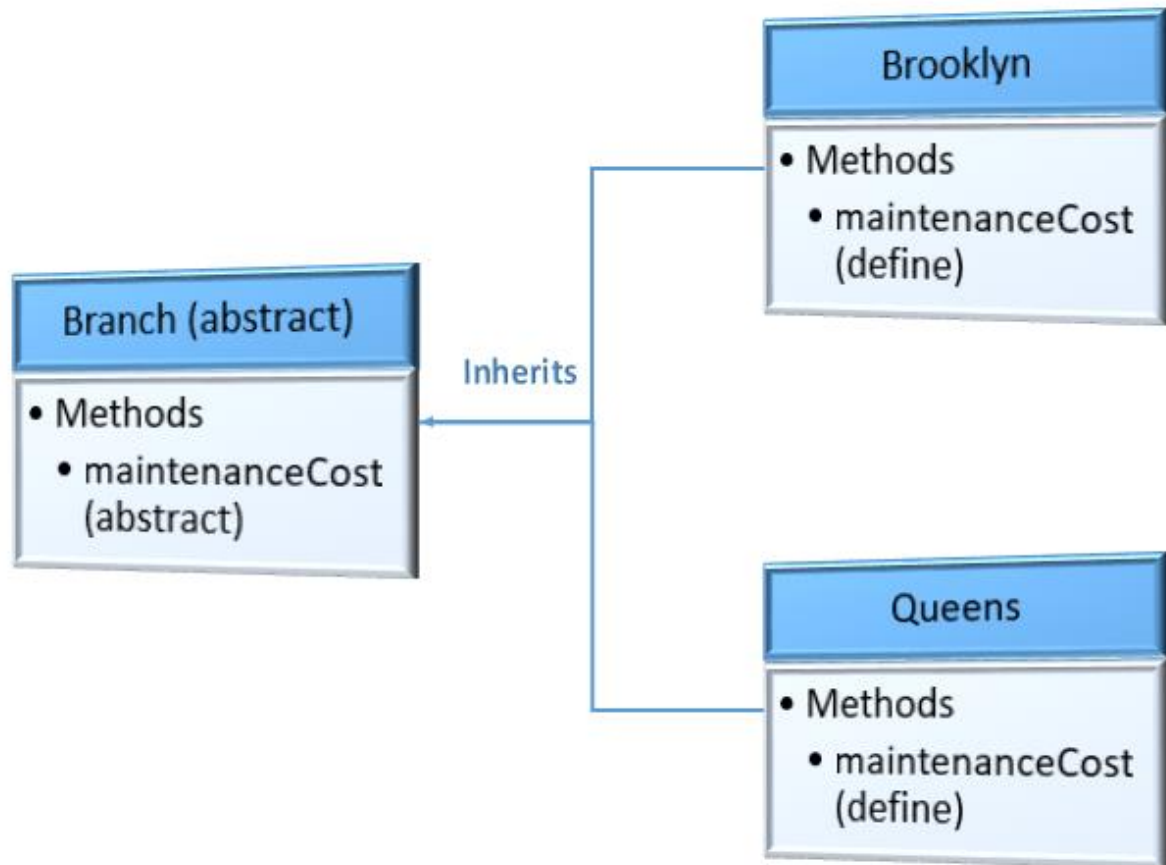


TypeError Traceback (most recent call last)
<ipython-input-46-349a6b0e68bd> in <module>
----> 1 objectAlbany = Branch()

TypeError: __init__() missing 5 required positional arguments: 'branchID', 'branchStreet', 'branchCity', 'branchState', and 'branchZip'

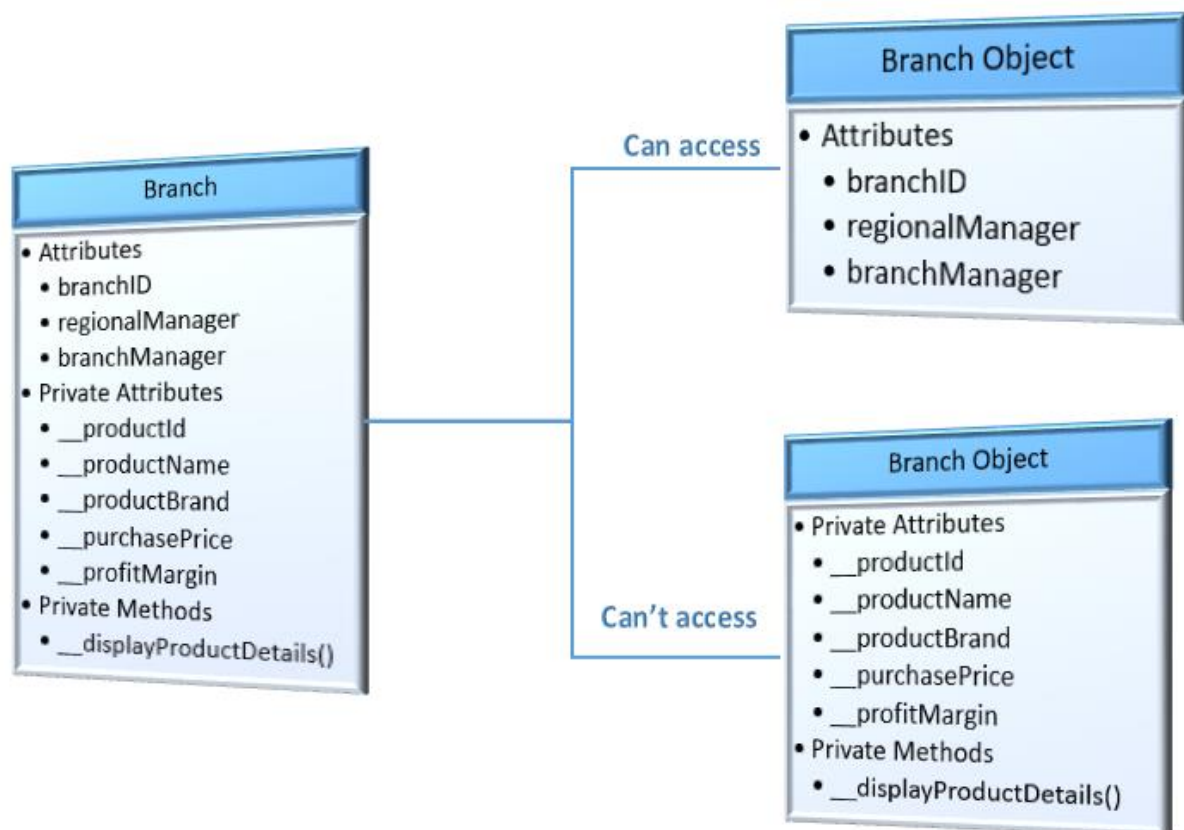






```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-52-8546e15cc93e> in <module>  
----> 1 branch = Branch()  
      2 branch.maintenanceCost()
```

TypeError: Can't instantiate abstract class Branch with abstract methods maintenanceCost



```

AttributeError                                Traceback (most recent call last)
<ipython-input-48-23ab3c57152f> in <module>
----> 1 branch.__profitMargin

```

AttributeError: 'Branch' object has no attribute '__profitMargin'

```

AttributeError                                Traceback (most recent call last)
<ipython-input-40-c62854f9e17a> in <module>
----> 1 branch.__displayProductDetails()

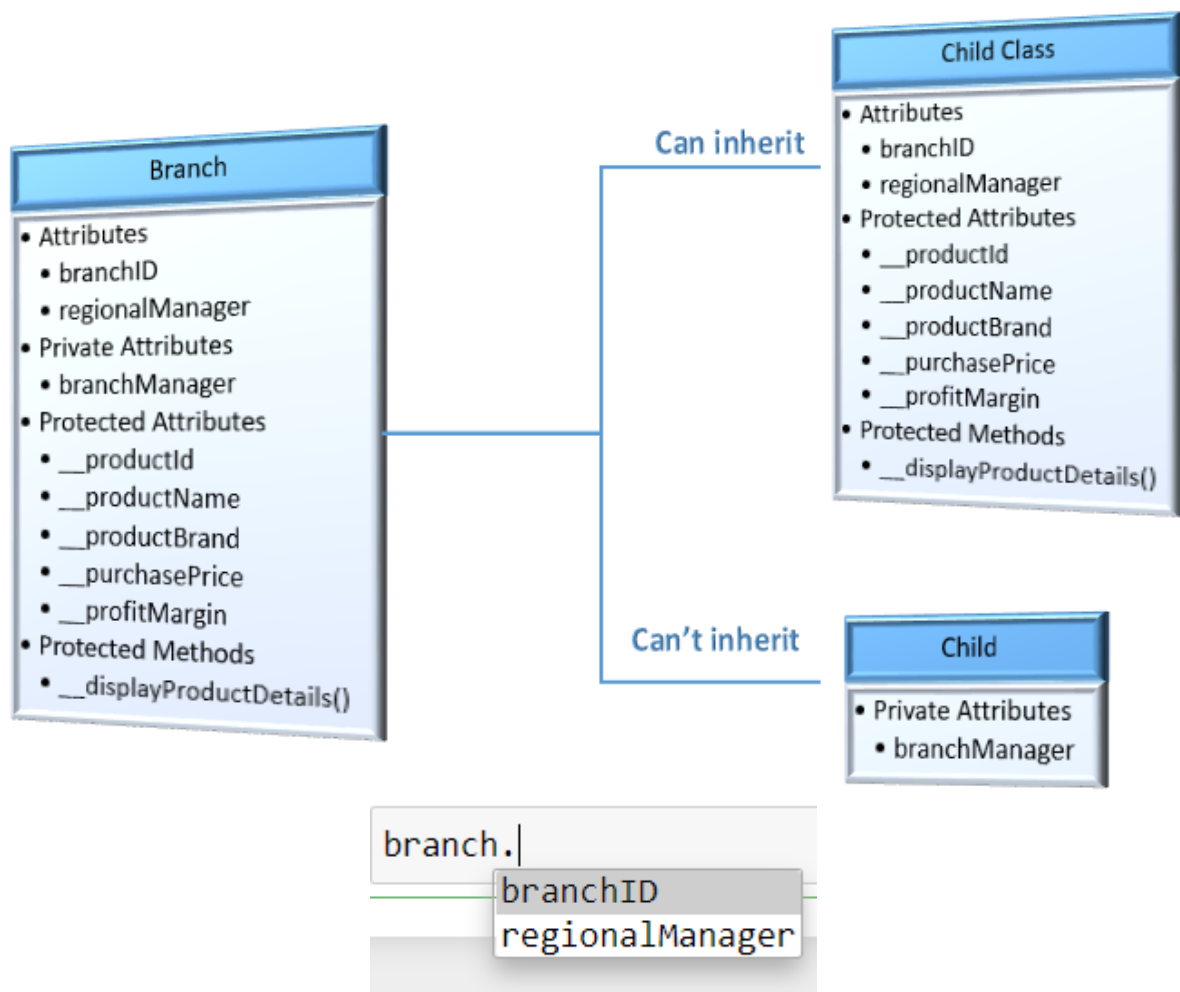
```

AttributeError: 'Branch' object has no attribute '__displayProductDetails'

```

branch.|
        branchID
        branchManager
        regionalManager

```



```

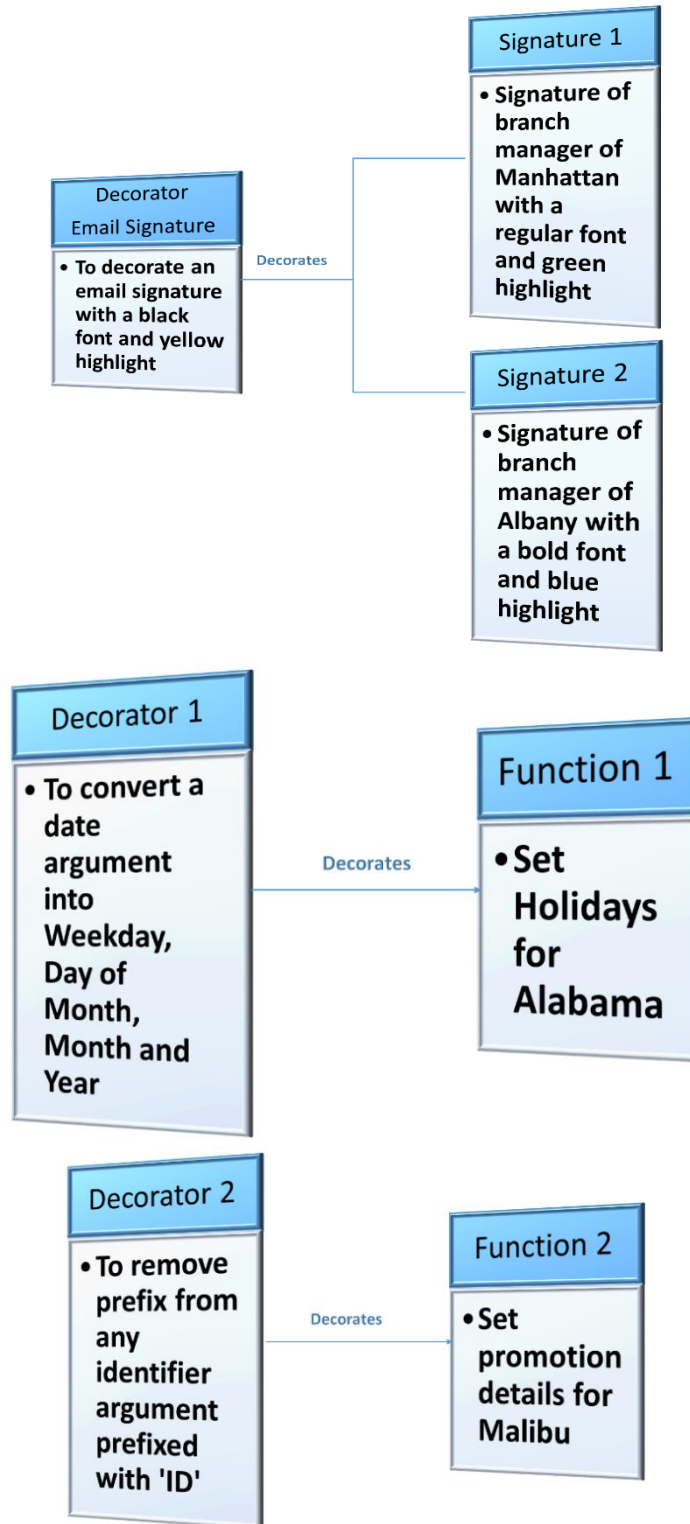
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-88-7c2f2e3d9a82> in <module>
----> 1 branchBrooklyn = Brooklyn()

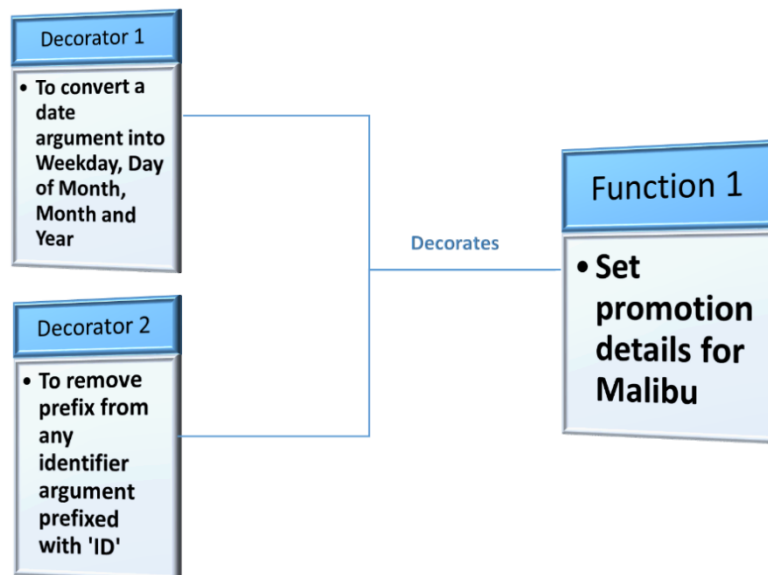
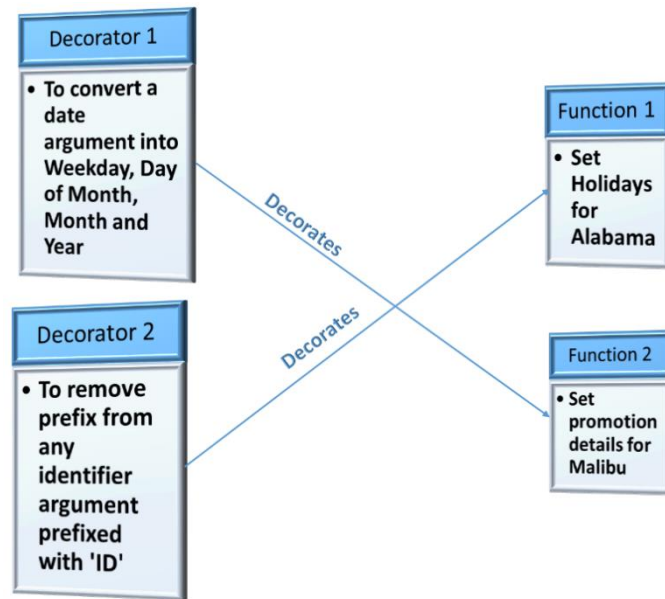
<ipython-input-87-a795a9221f51> in __init__(self)
      3     print(self._productId)
      4     self._displayProductDetails()
----> 5     print(self.__branchManager)

AttributeError: 'Brooklyn' object has no attribute '_Brooklyn__branchManager'

```

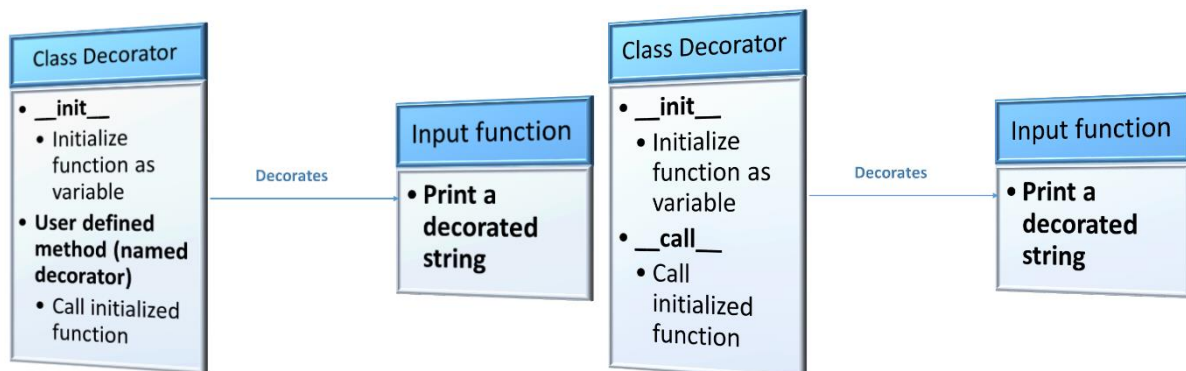
Chapter 3: Understanding Decorators and Their Applications

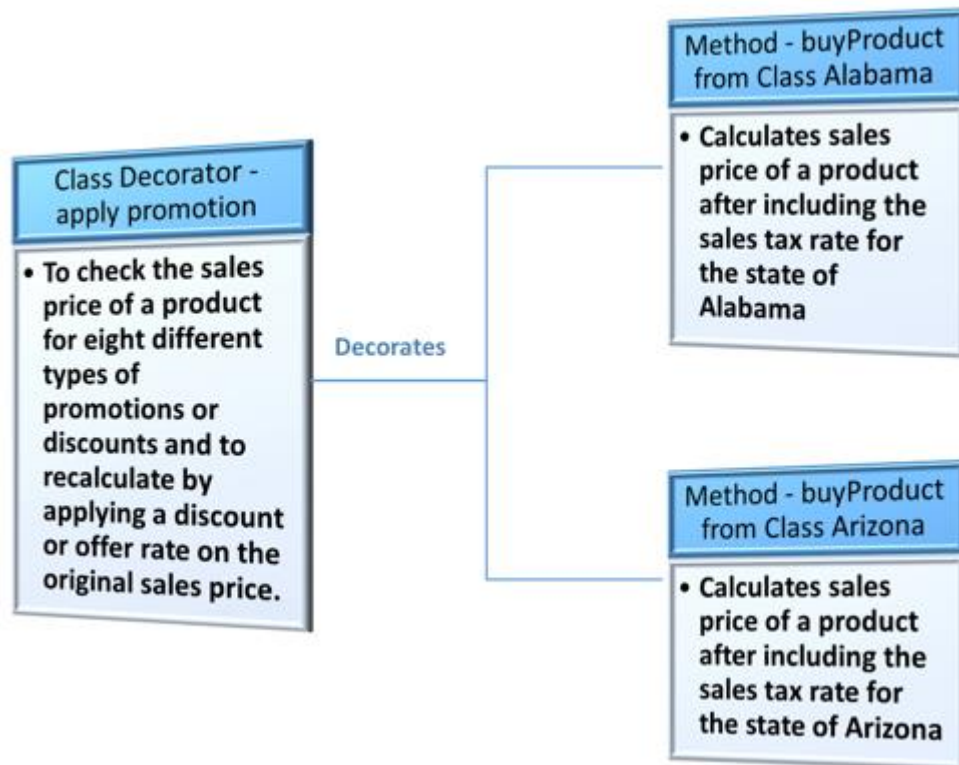




TypeError Traceback (most recent call last)
 <ipython-input-6-769efba848f3> in <module>
 ----> 1 inputfunction()

TypeError: 'classdecorator' object is not callable





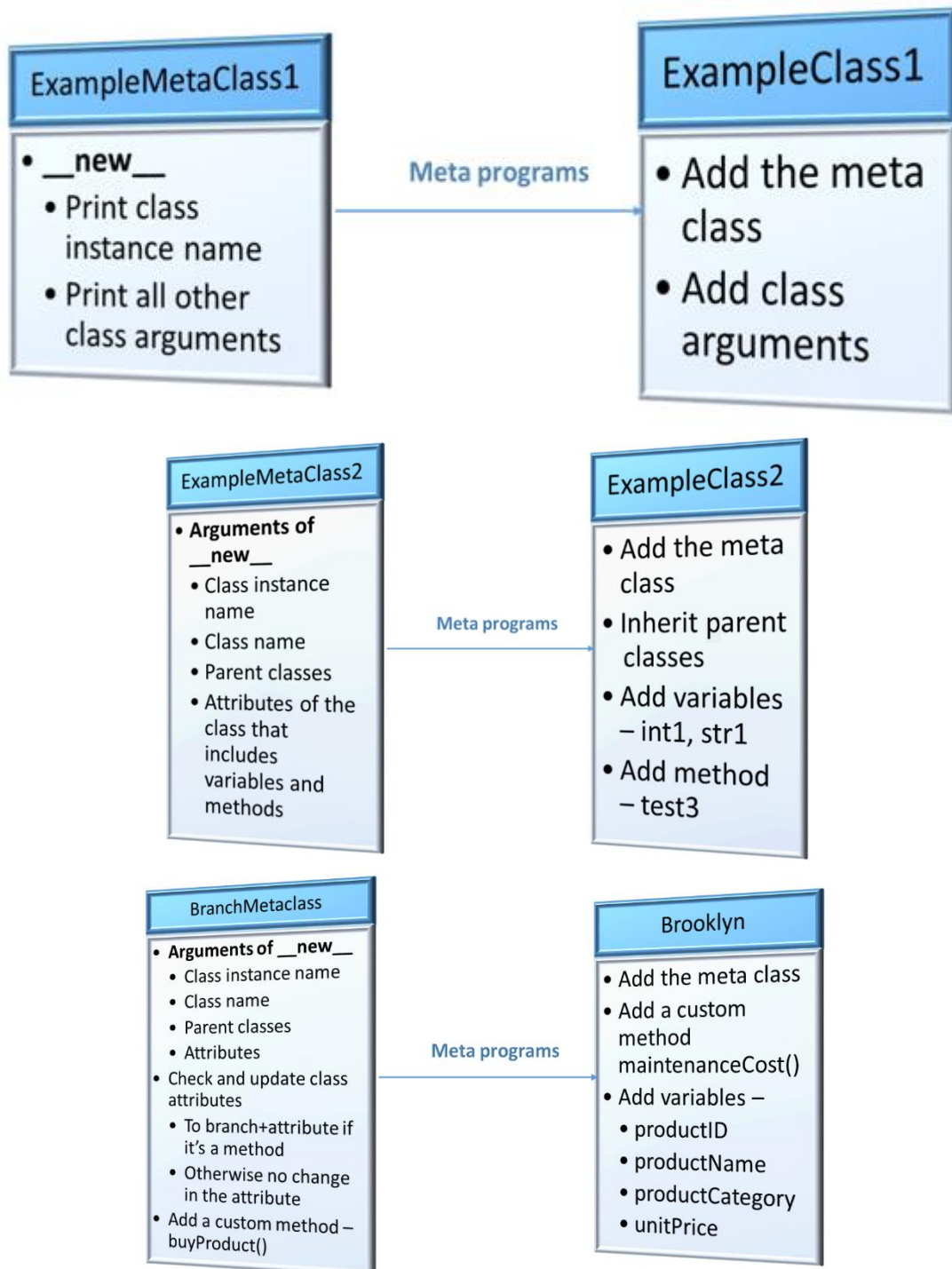
```
TypeError                                Traceback (most recent call last)
<ipython-input-351-32f04240a8b7> in <module>
----> 1 alb.buyProduct('Samsung-Refrigerator',200,1,'20%Off')
```

TypeError: buyProduct() takes 4 positional arguments but 5 were given

```
TypeError                                Traceback (most recent call last)
<ipython-input-347-cd717ea651b5> in <module>
----> 1 Alabama.anotherMethod()
```

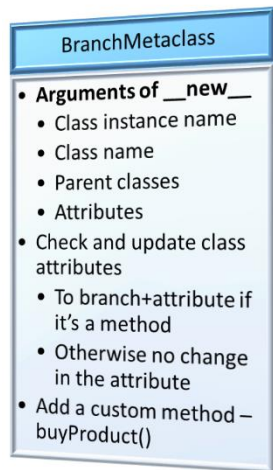
TypeError: anotherMethod() missing 1 required positional argument: 'self'

Chapter 4: Working with Metaclasses



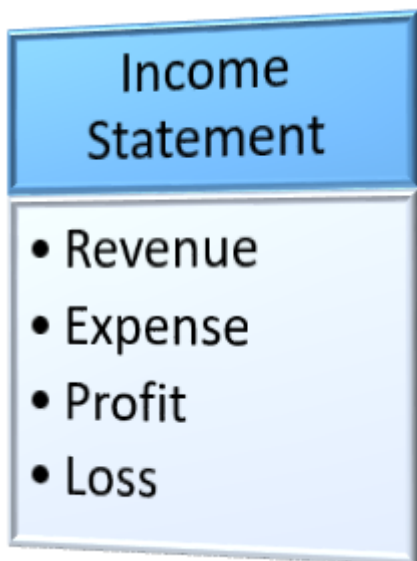
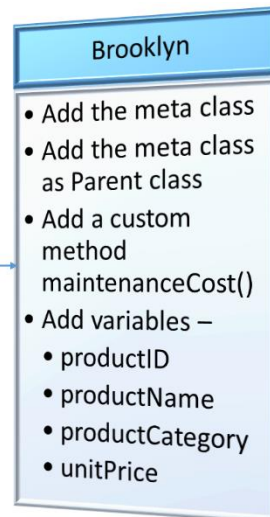
```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-214-bfc347d19820> in <module>  
----> 1 queens = Queens()
```

```
TypeError: __new__() missing 3 required positional arguments: 'classname', 'baseclasses', and 'attributes'
```



Meta programs

Inherits



Malibu.

```
assets
calc_expense
calc_revenue
calc_totalassets
calc_totalliabilities
liabilities
loss
mro
profit
```

Malibu.

```
assets
calc_totalassets
calc_totalliabilities
IncomeStatement_Calc_Expense
IncomeStatement_Calc_Revenue
IncomeStatement_Loss
IncomeStatement_Profit
liabilities
mro
```

Malibu.

assets
calc_totalassets
calc_totalliabilities
IncomeStatement_Calc_Expense
IncomeStatement_Calc_Revenue
IncomeStatement_Loss
IncomeStatement_Profit
liabilities
mro

Malibu.

BalanceSheet_Assets
BalanceSheet_Calc_Totalassets
BalanceSheet_Calc_Totalliabilities
BalanceSheet_Liabilities
calc_expense
calc_revenue
loss
mro
profit

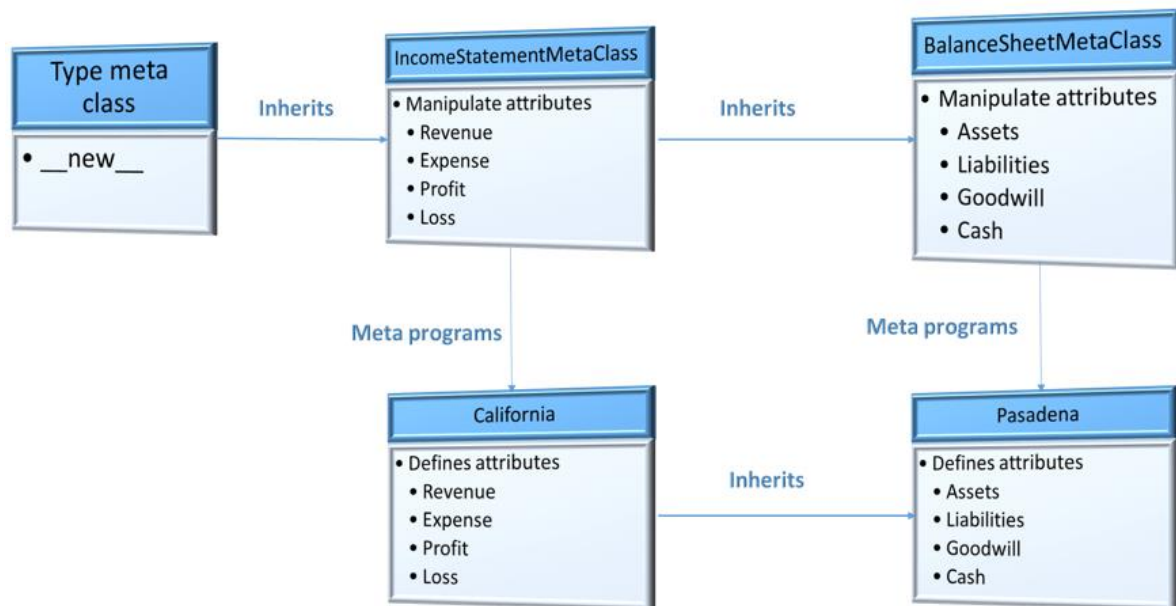
TypeError Traceback (most recent call last)

```
<ipython-input-47-a1dfffb78fdd6> in <module>
----> 1 class Pasadena(California,metaclass = BalanceSheetMetaClass):
      2     assets = 18000
      3     liabilities = 5000
      4     def calc_totalassets(cash,inventory,accountsreceivable):
      5         totalassets = cash + inventory + accountsreceivable
```

TypeError: metaclass conflict: the metaclass of a derived class must be a (non-strict) subclass of the metaclasses of all its bases

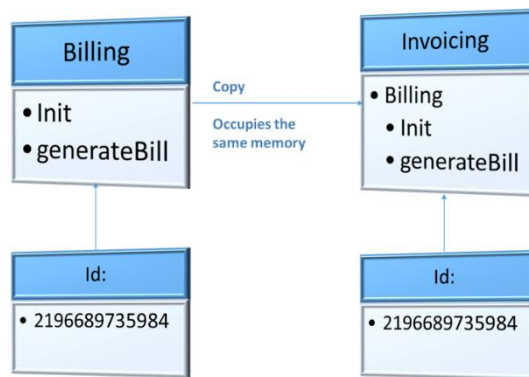
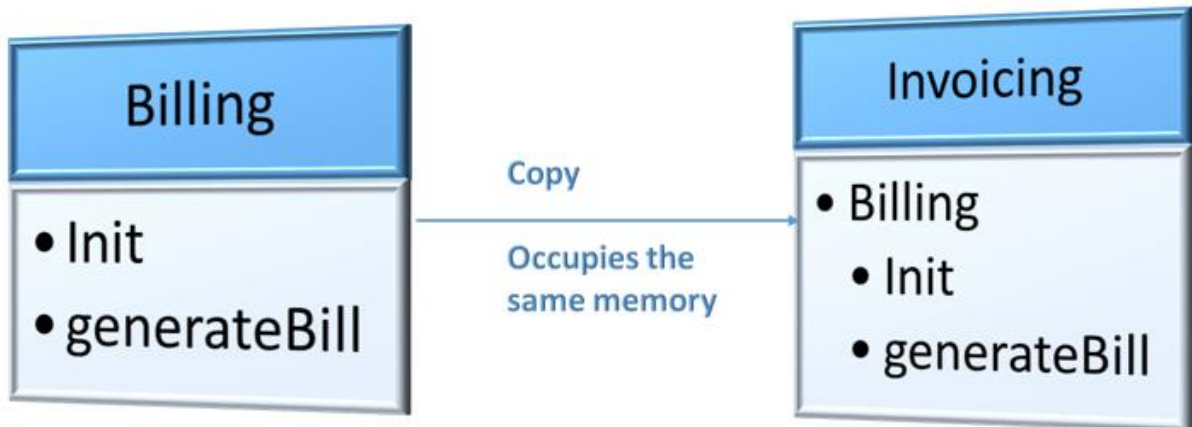
Pasadena.

BalanceSheet_Assets
BalanceSheet_Calc_Totalassets
BalanceSheet_Calc_Totalliabilities
BalanceSheet_Liabilities
IncomeStatement_Calc_Expense
IncomeStatement_Calc_Revenue
IncomeStatement_Loss
IncomeStatement_Profit
mro



	ColumnName	Type	Length
0	Productid	NUMERIC	6.0
1	Productname	VARCHAR	6.0
2	Productcategory	VARCHAR	11.0
3	Salesquantity	NUMERIC	1.0
4	Taxrate	NUMERIC	4.0
5	Salesprice	NUMERIC	4.0
6	Profit	NUMERIC	2.0
7	Loss	NUMERIC	1.0
8	Salesmargin	NUMERIC	3.0
9	Promotion	VARCHAR	6.0
10	Promotionreason	VARCHAR	8.0
11	Instock	BOOLEAN	NaN

Chapter 5: Understanding Introspection



	ProductID	ProductName	Date	UnitPrice	Quantity	UnitDiscount	Tax	Amount Before Tax	Amount After Tax
0	300021	Samsung-Refrigerator	08/04/2021	200	25	10	0.0522	4750	4750.0522
	ProductID	ProductName	Date	UnitPrice	Quantity	UnitDiscount	Tax	Amount Before Tax	Amount After Tax
0	100011	Delmont Ketchup	09/04/2021	5	0.25	0.1	0.0522	1.225	1.2772
	ProductID	ProductName	Date	UnitPrice	Quantity	UnitDiscount	Tax	Amount Before Tax	Amount After Tax
0	300021	Samsung-Refrigerator	08/04/2021	200	25	10	0.0522	4750	4750.0522

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-88-36f88e397ab9> in <module>  
----> 1 inventory.returnInventory()
```

```
AttributeError: 'str' object has no attribute 'returnInventory'
```

	ProductID	Date	UnitPrice	Quantity	UnitDiscount	Tax	Amount Before Tax	Amount Before Tax	SupplierDetails
0	100011	09/04/2021	5	0.25	0.1	0.0522	1.225	1.2772	{'SupplierName': 'Test Supplier', 'SupplierCod...

get testAttr

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-148-b983341ece3a> in <module>
----> 1 testPropertyObject.testAttr
```

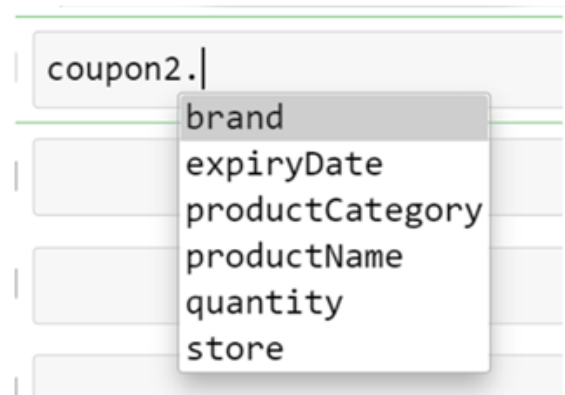
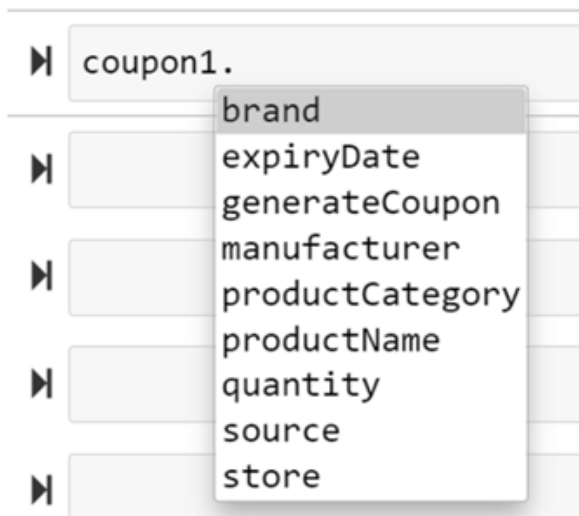
```
<ipython-input-142-ad7fa9f7c228> in get_testAttr(self)
      5     def get_testAttr(self):
      6         print("get testAttr")
----> 7         return self._testAttr
      8
      9     def set_testAttr(self, value):
```

```
AttributeError: 'testPropertyClass' object has no attribute '_testAttr'
```

Chapter 6: Implementing Reflection on Python Objects

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-11-1b826290f64a> in <module>  
----> 1 Coupon1
```

NameError: name 'Coupon1' is not defined



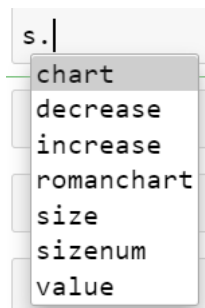
Chapter 7: Understanding Generics and Typing

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-6-fa1ce8e899e3> in <module>  
----> 1 fashion.getitem()  
  
<ipython-input-4-796edca1116a> in getitem(self)  
    32  
    33     def getitem(self):  
--> 34         return self.clothingCategory,self.gender,self.model,self.design,self.price, self.size,self.color  
  
AttributeError: 'Fashion' object has no attribute 'clothingCategory'
```

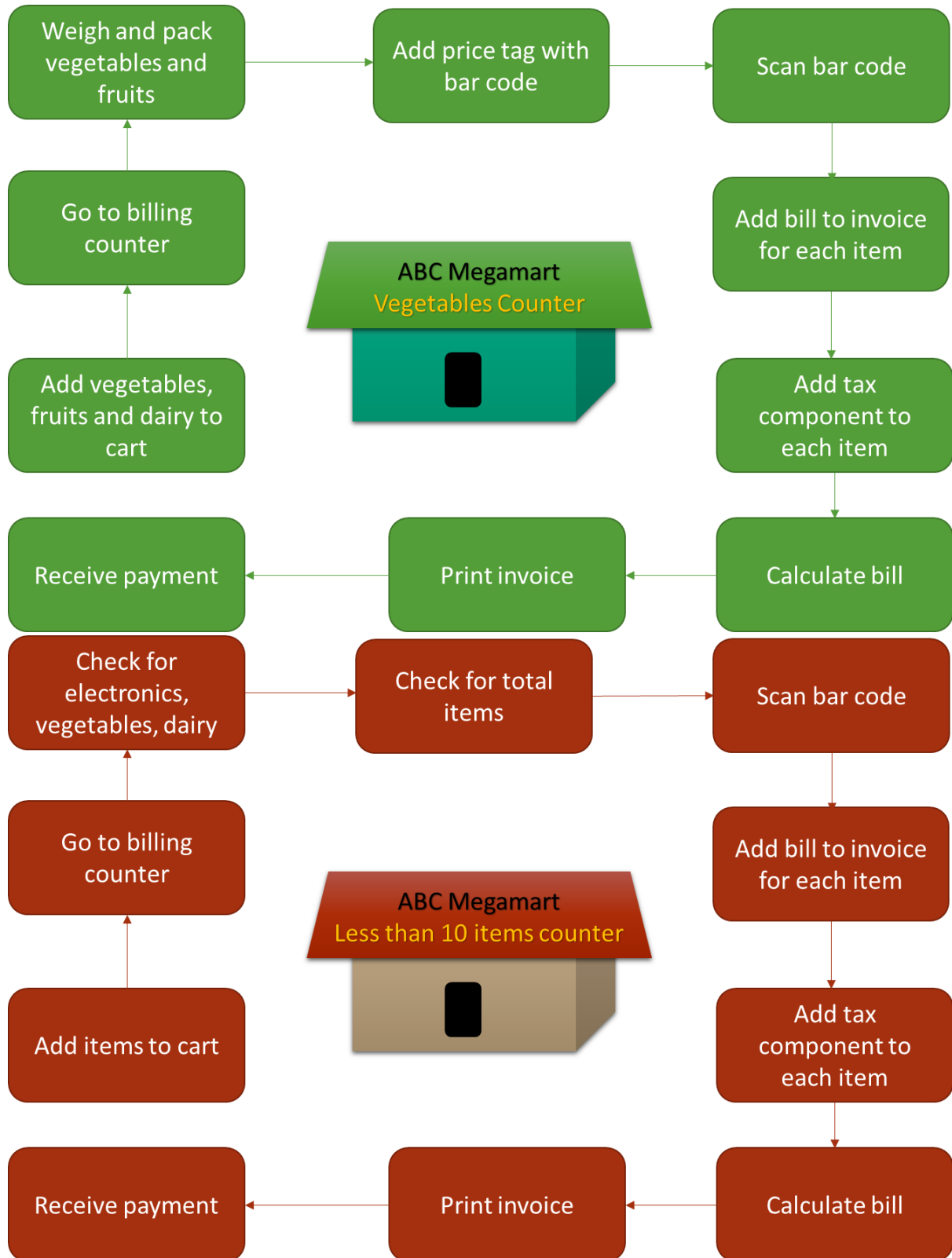
Size should be an integer

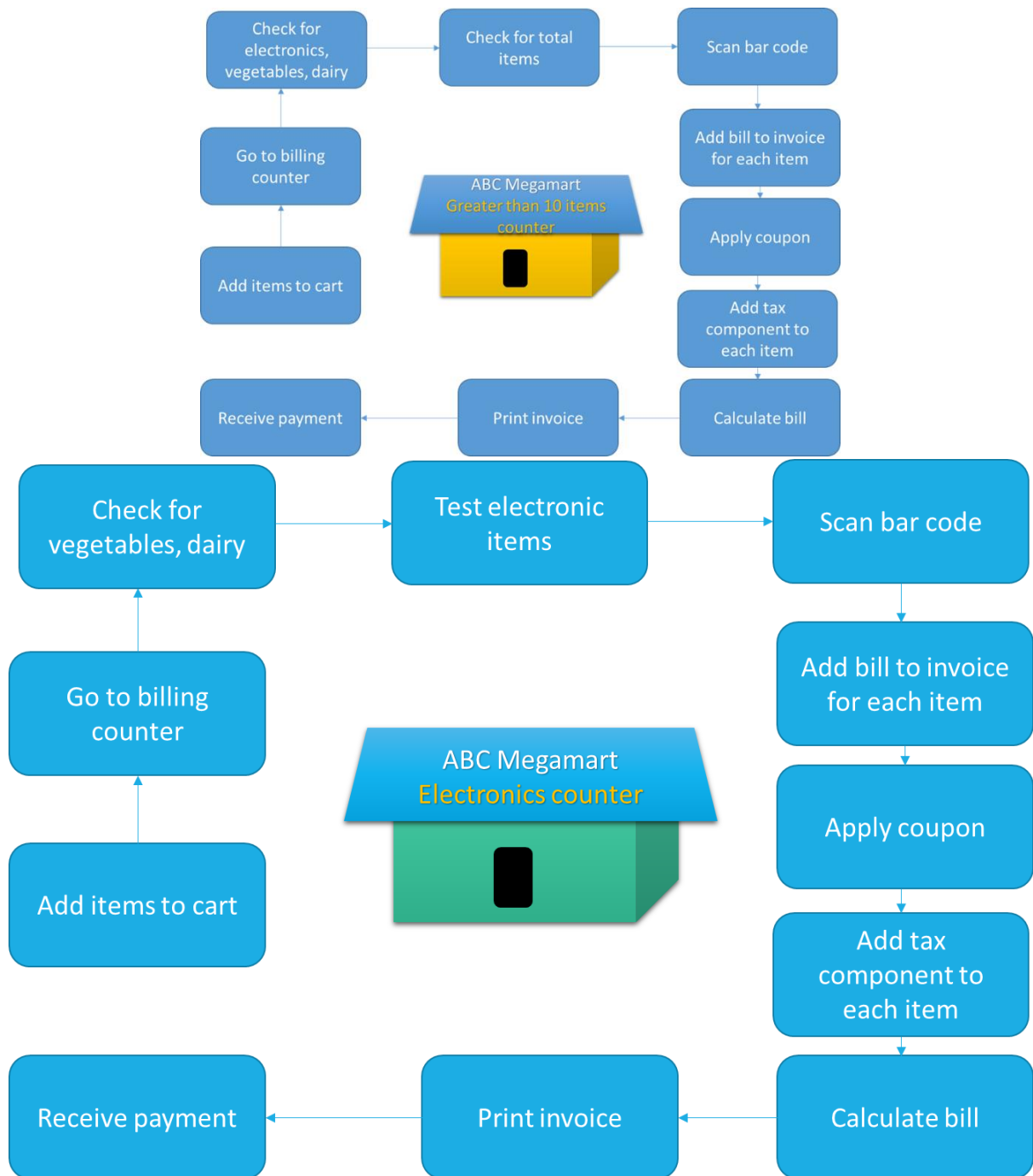
Size should be a valid dress size

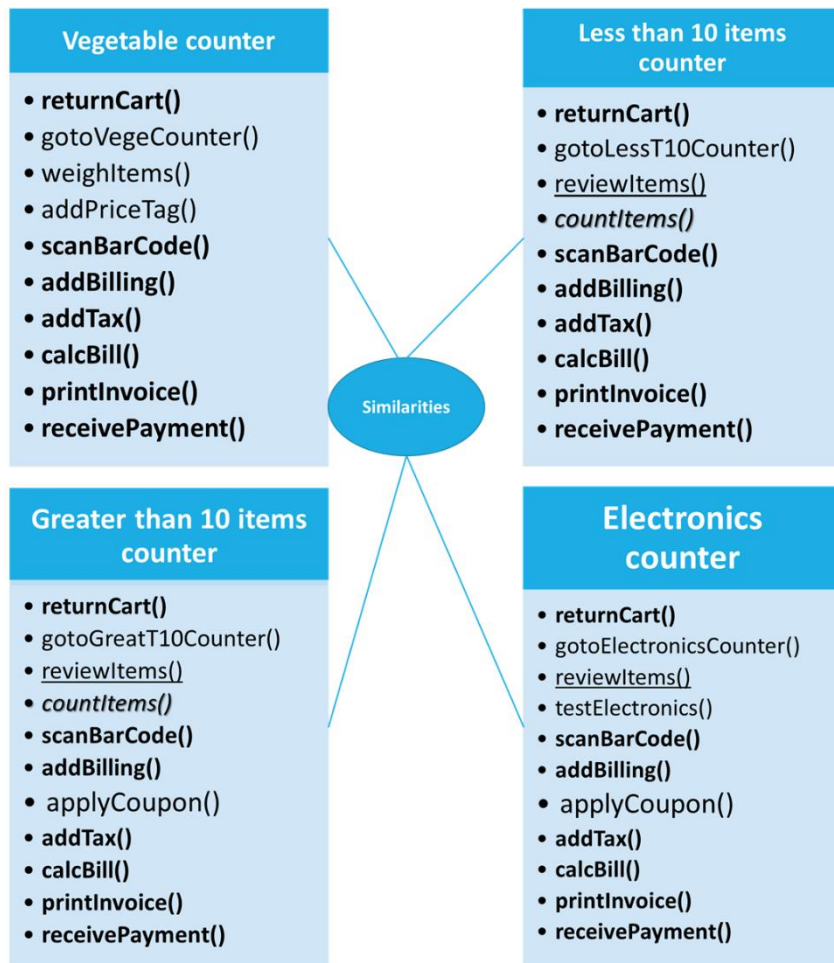
```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-53-3d2db1bf7adf> in <module>  
----> 1 s.value()  
  
<ipython-input-51-5269406ba653> in value(self)  
    10  
    11     def value(self):  
--> 12         return self.size  
    13  
    14     def __add__(self,second):  
  
AttributeError: 'DressSize' object has no attribute 'size'
```



Chapter 8: Defining Templates for Algorithms

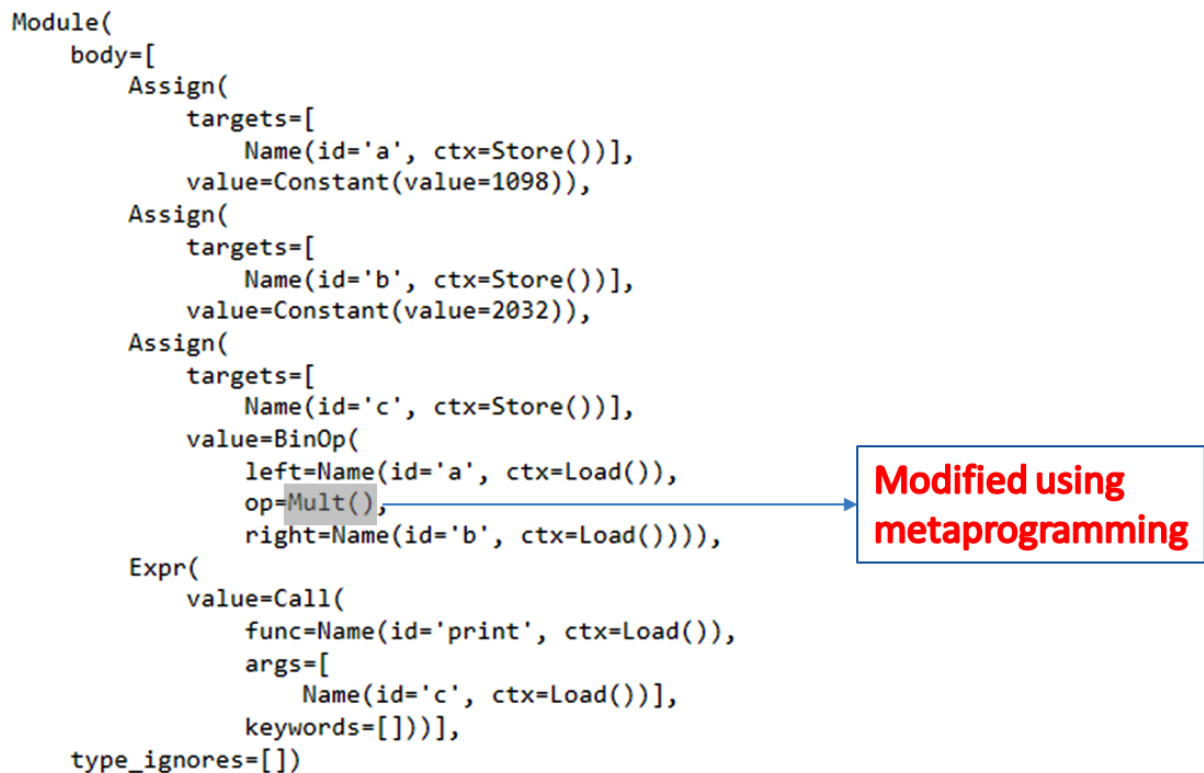
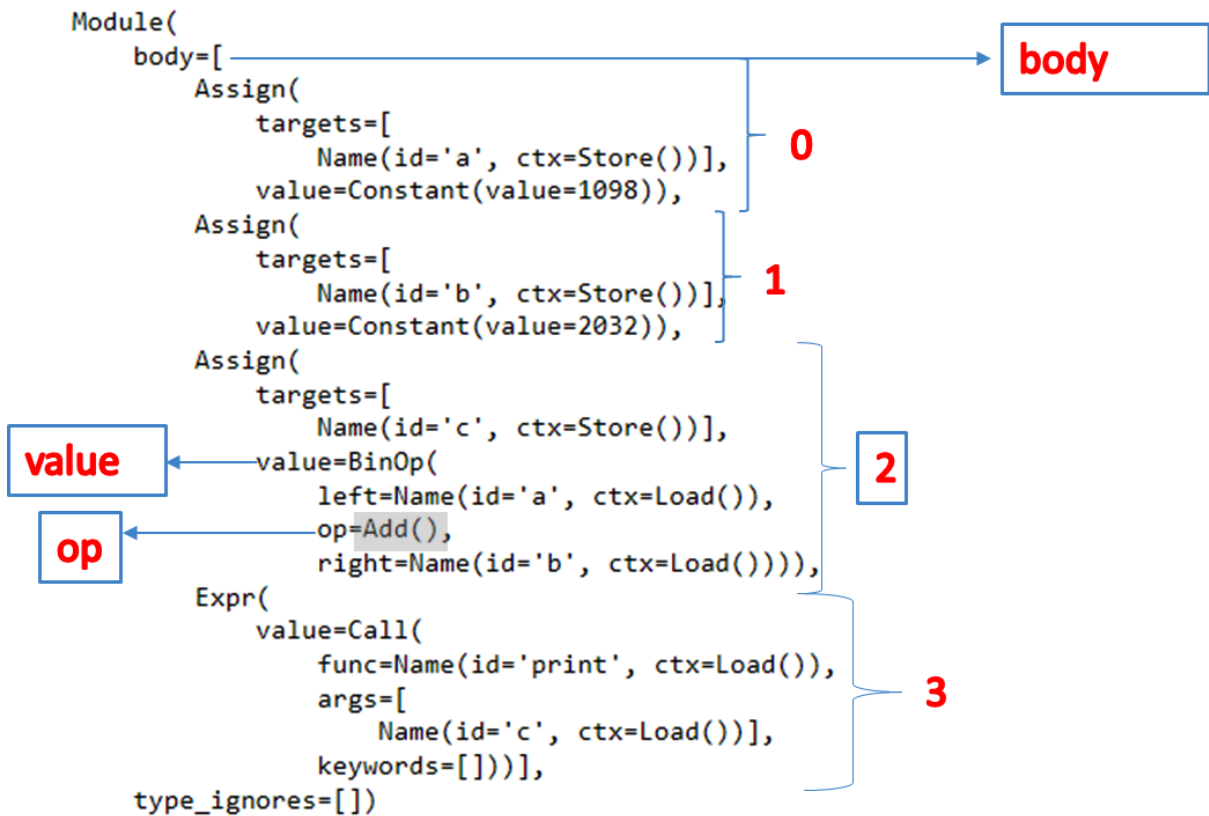






Chapter 9: Understanding Code through Abstract Syntax Tree

```
Module(  
  body=[  
    Assign(  
      targets=[  
        Name(id='productName', ctx=Store())],  
      value=Constant(value='Iphone X')),  
    type_ignores=[])  
  Module(  
    body=[  
      Assign(  
        targets=[  
          Name(id='a', ctx=Store())],  
        value=Constant(value=1098)),  
      Assign(  
        targets=[  
          Name(id='b', ctx=Store())],  
        value=Constant(value=2032)),  
      Assign(  
        targets=[  
          Name(id='c', ctx=Store())],  
        value=BinOp(  
          left=Name(id='a', ctx=Load()),  
          op=Add(),  
          right=Name(id='b', ctx=Load()))),  
      Expr(  
        value=Call(  
          func=Name(id='print', ctx=Load()),  
          args=[  
            Name(id='c', ctx=Load())],  
          keywords=[])),  
    type_ignores=[])
```



```

Module(
  body=[
    ClassDef(
      name='Branch',
      bases=[],
      keywords=[],
      body=[
        Expr(
          value=Constant(value='attributes...')),
        Expr(
          value=Constant(value='methods...'))],
      decorator_list=[]),
    type_ignores=[])
Module(
  body=[
    ClassDef(
      name='Branch',
      bases=[],
      keywords=[],
      body=[
        Assign(
          targets=[
            Name(id='branchID', ctx=Store())],
          value=Constant(value=1001)),
        FunctionDef(
          name='getProduct',
          args=arguments(
            posonlyargs=[],
            args=[
              arg(arg='self')],
            kwonlyargs=[],
            kw_defaults=[],
            defaults=[]),
          body=[
            Return(
              value=Constant(value='product'))],
          decorator_list=[
            Name(id='staticmethod', ctx=Load())]),
        decorator_list=[]],
    type_ignores=[])

```

```

Module(
  body=[
    Assign(
      targets=[
        Name(id='branchAlbany', ctx=Store())],
      value=Call(
        func=Name(id='Branch', ctx=Load()),
        args=[],
        keywords=[])),
    type_ignores=[]))
Module(
  body=[
    ClassDef(
      name='VegCounter',
      bases=[],
      keywords=[],
      body=[
        FunctionDef(
          name='returnCart',
          args=arguments(
            posonlyargs=[],
            args=[
              arg(arg='self')],
            vararg=arg(arg='items'),
            kwonlyargs=[],
            kw_defaults=[],
            defaults=[]),
          body=[
            Assign(
              targets=[
                Name(id='cartItems', ctx=Store())],
              value=List(elts=[], ctx=Load())),
            For(
              target=Name(id='i', ctx=Store()),
              iter=Name(id='items', ctx=Load()),
              body=[
                Expr(
                  value=Call(
                    func=Attribute(
                      value=Name(id='cartItems', ctx=Load()),
                      attr='append',
                      ctx=Load()),
                    args=[
                      Name(id='i', ctx=Load())],
                    keywords=[])),
                or_else=[]),
              Return(
                value=Name(id='cartItems', ctx=Load()))],
              decorator_list=[])),
    decorator_list=[]),
  decorator_list=[]),

```

```

Assign(
  targets=[
    Name(id='veg', ctx=Store())],
  value=Call(
    func=Name(id='VegCounter', ctx=Load()),
    args=[],
    keywords=[])),
Expr(
  value=Call(
    func=Name(id='print', ctx=Load()),
    args=[
      Call(
        func=Attribute(
          value=Name(id='veg', ctx=Load()),
          attr='returnCart',
          ctx=Load()),
        args=[
          Constant(value='onions'),
          Constant(value='potatoes'),
          Constant(value='carrots'),
          Constant(value='lettuce')],
        keywords=[])],
      keywords=[])),
  type_ignores=[])
Expr(
  value=Call(
    func=Name(id='print', ctx=Load()),
    args=[
      Call(
        func=Attribute(
          value=Name(id='veg', ctx=Load()),
          attr='returnCart',
          ctx=Load()),
        args=[
          Constant(value='onions'),
          Constant(value='potatoes'),
          Constant(value='carrots'),
          Constant(value='lettuce')],
        keywords=[])],
      keywords=[])),
  type_ignores=[])

```

```

Module(
  body=[
    Expr(
      value=Call(
        func=Attribute(
          value=Name(id='veg', ctx=Load()),
          attr='returnCart',
          ctx=Load()),
        args=[
          Constant(value='onions'),
          Constant(value='tomatoes'),
          Constant(value='carrots'),
          Constant(value='lettuce')],
        keywords=[])),
    type_ignores=[])

```

NodeTransformer.

generic_visit
mro
visit
visit_Constant

```

Module(
  body=[
    Expr(
      value=Call(
        func=Attribute(
          value=Name(id='veg', ctx=Load()),
          attr='returnCart',
          ctx=Load()),
        args=[
          Constant(value='item: onions'),
          Constant(value='item: tomatoes'),
          Constant(value='item: carrots'),
          Constant(value='item: lettuce')],
        keywords=[])),
    type_ignores=[])

```

Chapter 10: Understanding Method Resolution Order of Inheritance

```
-----  
TypeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_2164\409991840.py in <module>  
----> 1 class ScanCode(ElectronicsCounter, VegeCounter):  
      2     pass
```

```
TypeError: Cannot create a consistent method resolution  
order (MRO) for bases CommonCounter, CheckItems
```

Chapter 11: Creating Dynamic Objects

type()

```
Init signature: type(self, /, *args, **kwargs)
Docstring:
type(object_or_name, bases, dict)
type(object) -> the object's type
type(name, bases, dict) -> a new type
Type:
    type
Subclasses:
    ABCMeta, EnumMeta, NamedTupleMeta, _TypedDictMeta, _ABC, MetaHasDescripto
rs, PyCStructType, UnionType, PyCPointerType, PyCArrayType, ...
```

branchAlbany.

branchCity
branchID
branchState
branchStreet
branchZip
mro

Branch

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__']
```

branchAlbany

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__',  
 'branchCity',  
 'branchID',  
 'branchState',  
 'branchStreet',  
 'branchZip']
```

Branch

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__']
```

branchAlbany

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__',  
 'branchCity',  
 'branchID',  
 'branchState',  
 'branchStreet',  
 'branchZip']
```

branchNYC

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__',  
 'branchID',  
 'productBrand',  
 'productId',  
 'productName']
```

Product

```
[ '__class__',  
  '__delattr__',  
  '__dict__',  
  '__dir__',  
  '__doc__',  
  '__eq__',  
  '__format__',  
  '__ge__',  
  '__getattr__',  
  '__gt__',  
  '__hash__',  
  '__init__',  
  '__init_subclass__',  
  '__le__',  
  '__lt__',  
  '__module__',  
  '__ne__',  
  '__new__',  
  '__reduce__',  
  '__reduce_ex__',  
  '__repr__',  
  '__setattr__',  
  '__sizeof__',  
  '__str__',  
  '__subclasshook__',  
  '__weakref__']
```

product1

```
['ProductBrand',  
'ProductID',  
'ProductName',  
'ProfitMargin',  
'PurchasePrice',  
__class__,  
__delattr__,  
__dict__,  
__dir__,  
__doc__,  
__eq__,  
__format__,  
__ge__,  
__getattribute__,  
__gt__,  
__hash__,  
__init__,  
__init_subclass__,  
__le__,  
__lt__,  
__module__,  
__ne__,  
__new__,  
__reduce__,  
__reduce_ex__,  
__repr__,  
__setattr__,  
__sizeof__,  
__str__,  
__subclasshook__,  
__weakref__,  
branchID]
```

product2

```
['__class__',  
__delattr__,  
__dict__,  
__dir__,  
__doc__,  
__eq__,  
__format__,  
__ge__,  
__getattribute__,  
__gt__,  
__hash__,  
__init__,  
__init_subclass__,  
__le__,  
__lt__,  
__module__,  
__ne__,  
__new__,  
__reduce__,  
__reduce_ex__,  
__repr__,  
__setattr__,  
__sizeof__,  
__str__,  
__subclasshook__,  
__weakref__,  
productCategory',  
productID',  
productName',  
unitPrice']
```

product3

```
['Product Brand',  
'Product ID',  
'Product Name',  
'Profit Margin',  
'Purchase Price',  
__class__,  
__delattr__,  
__dict__,  
__dir__,  
__doc__,  
__eq__,  
__format__,  
__ge__,  
__getattribute__,  
__gt__,  
__hash__,  
__init__,  
__init_subclass__,  
__le__,  
__lt__,  
__module__,  
__ne__,  
__new__,  
__reduce__,  
__reduce_ex__,  
__repr__,  
__setattr__,  
__sizeof__,  
__str__,  
__subclasshook__,  
__weakref__,  
branchBorough',  
branchCity',  
branchID',  
branchStreet']
```

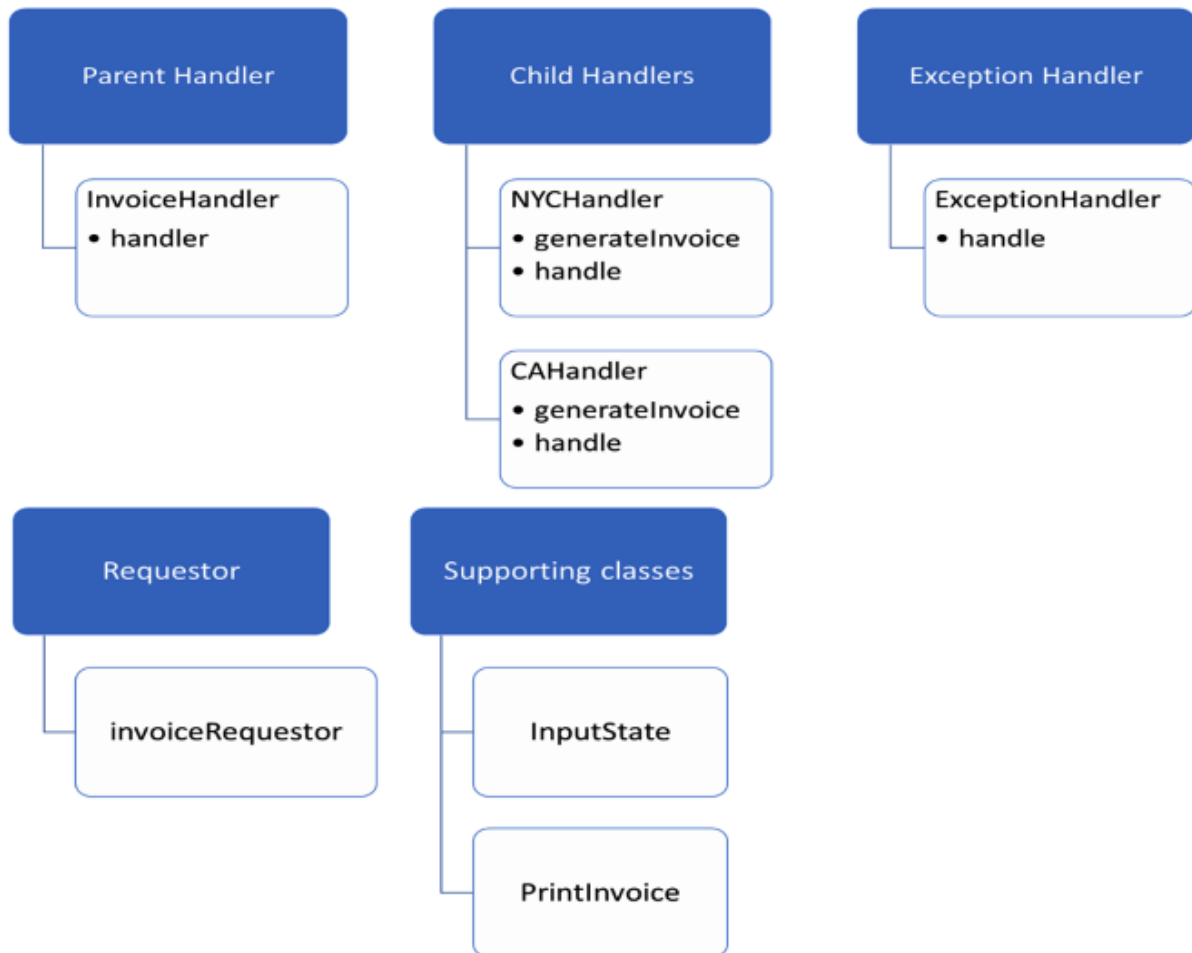
SimpleCoupon

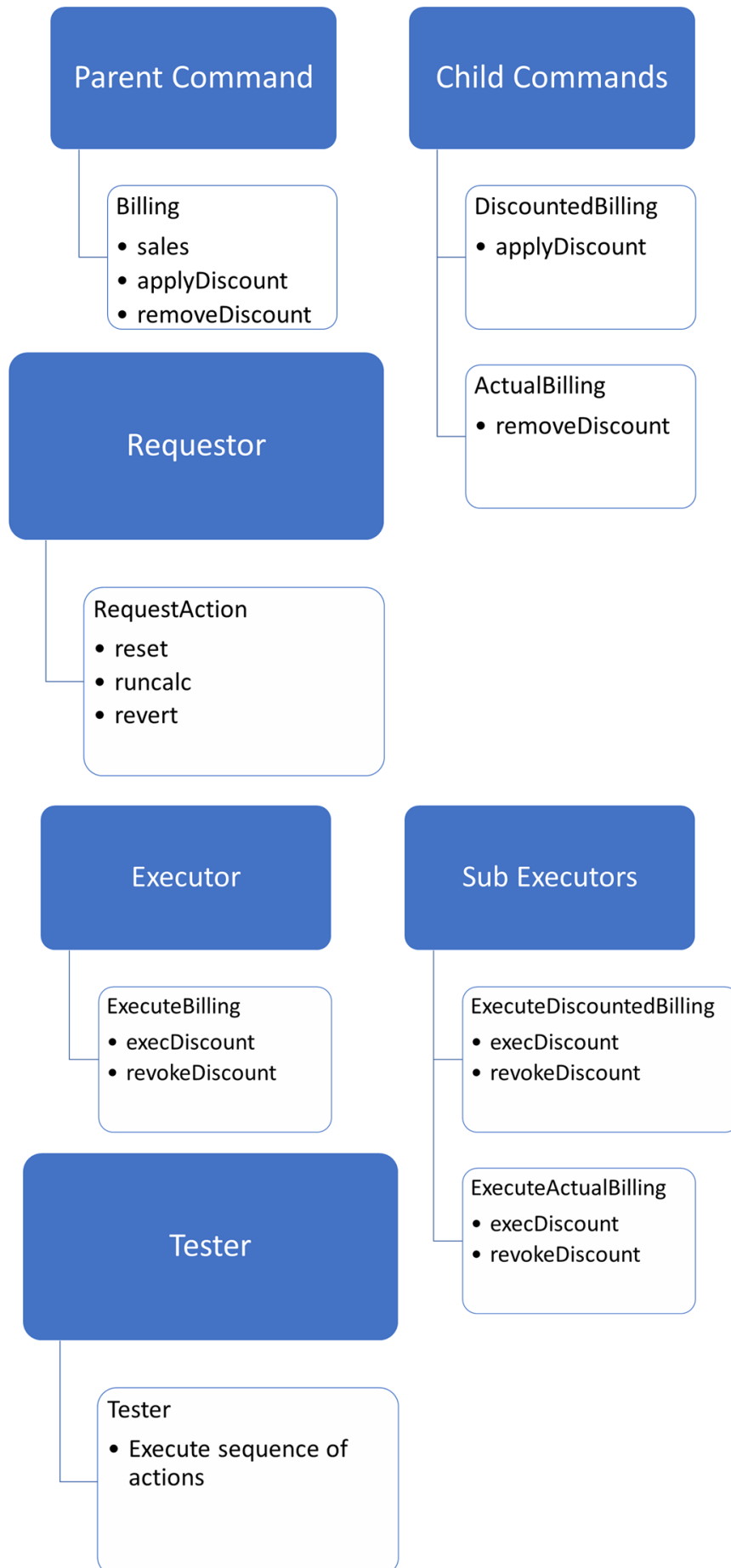
SimpleCoupon

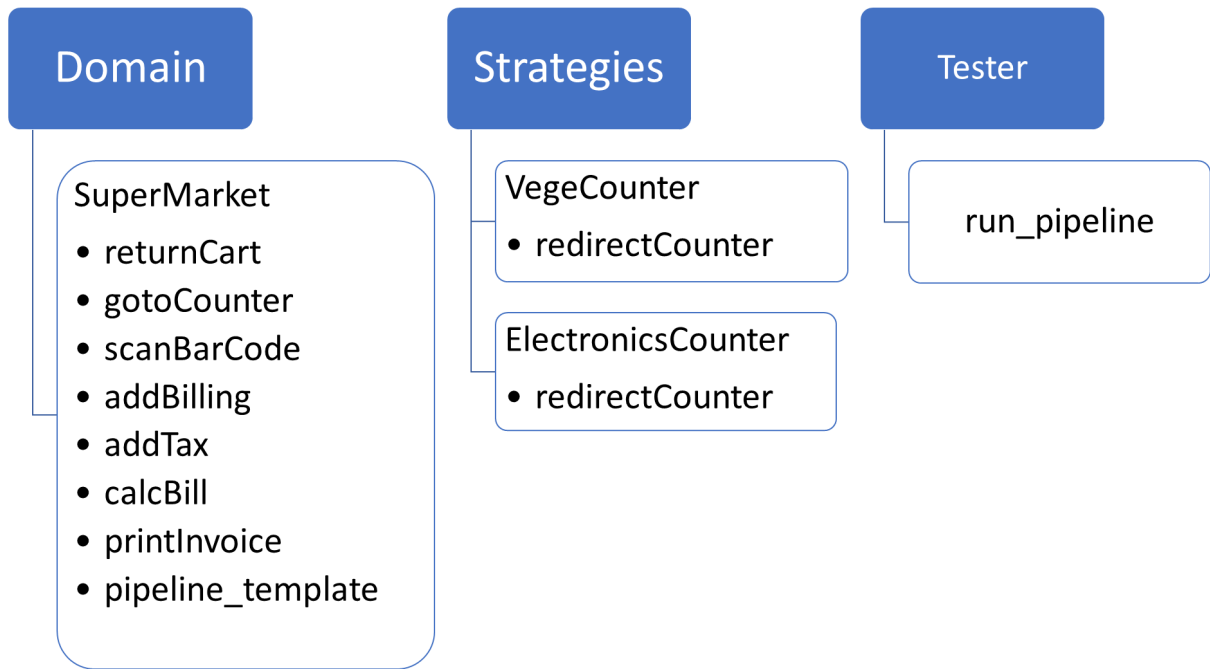
```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattr__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__',  
 'couponDetails']
```

```
['__class__',  
 '__delattr__',  
 '__dict__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattr__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__le__',  
 '__lt__',  
 '__module__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 '__weakref__',  
 'couponDetails',  
 'generateCondimentsCoupon',  
 'generateDessertsCoupon',  
 'generateSnacksCoupon']
```

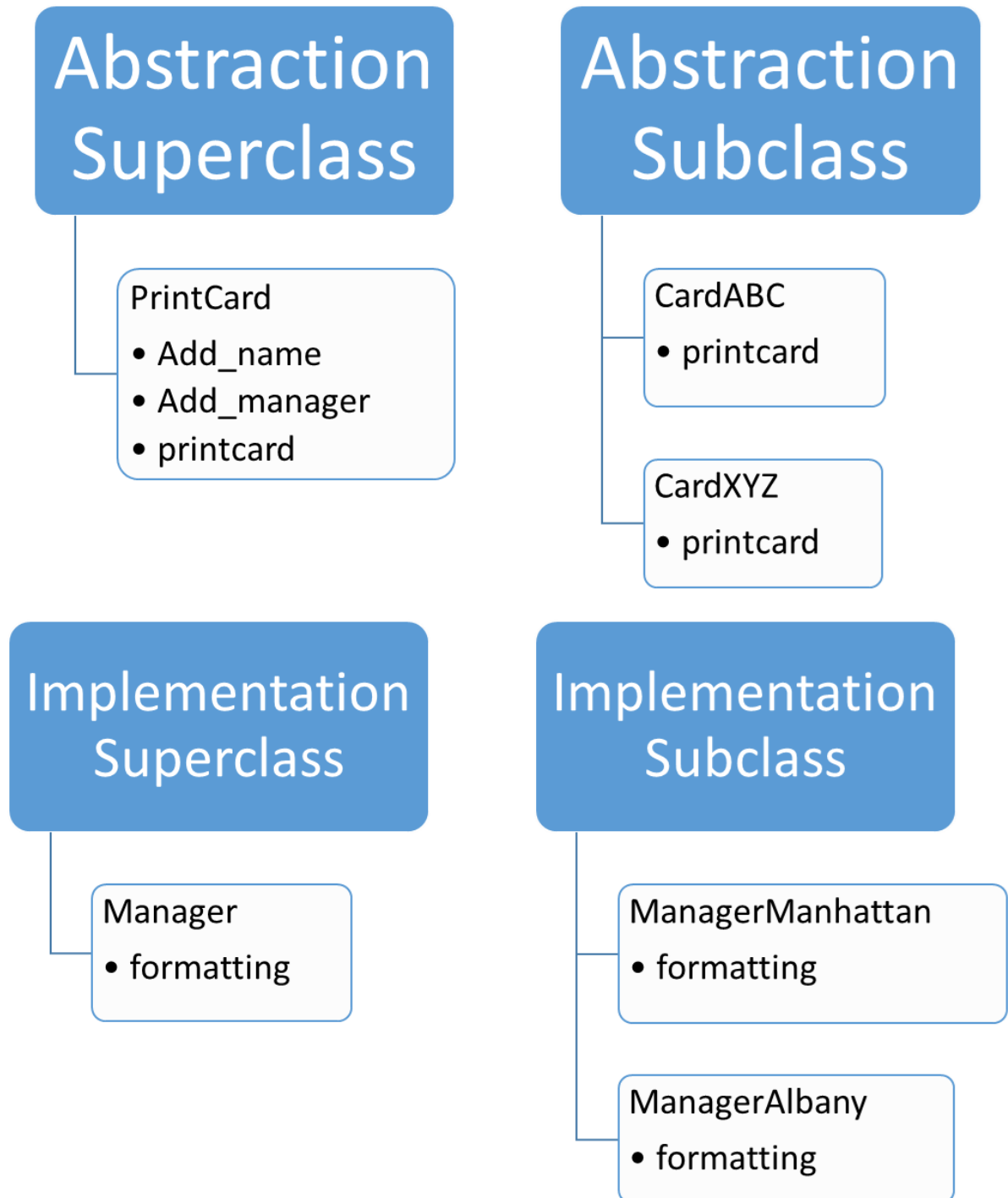
Chapter 12: Applying GOF Design Patterns – Part 1

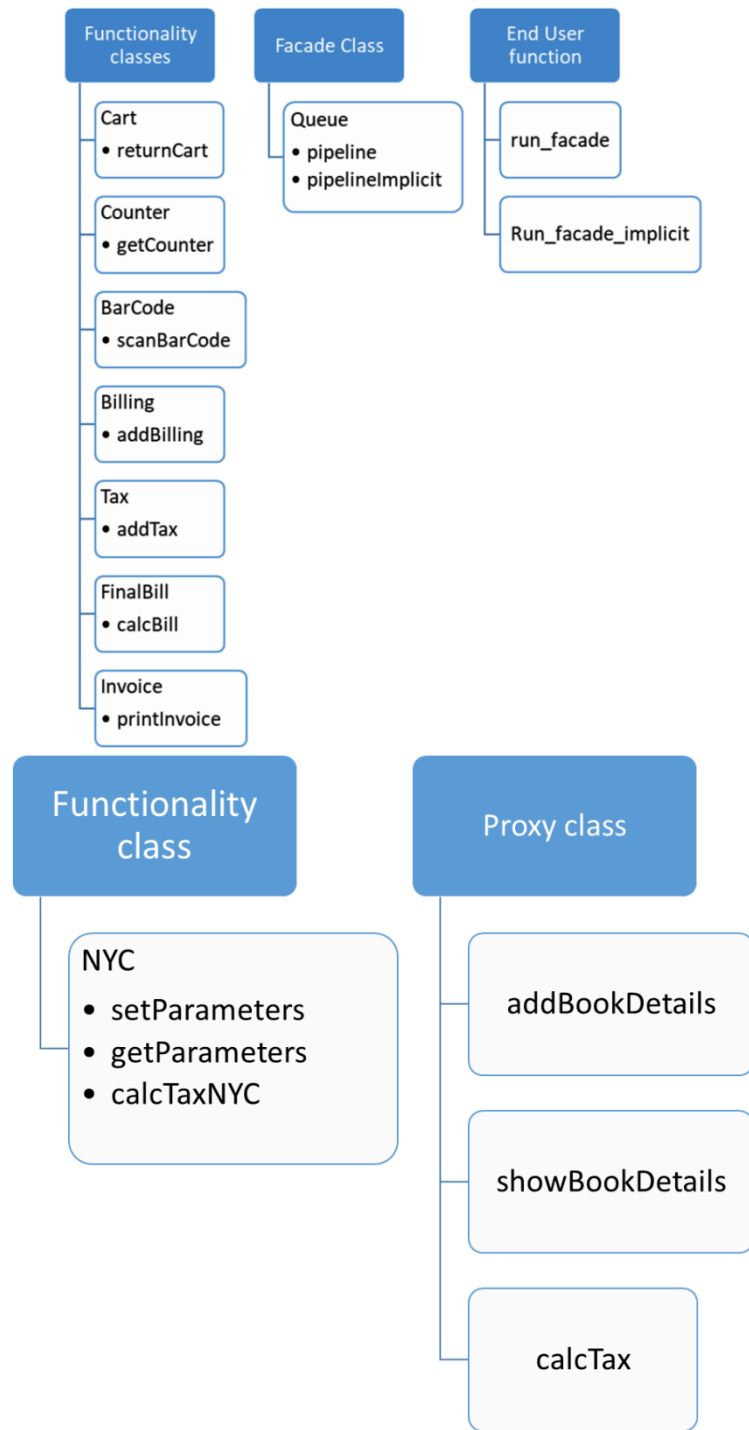


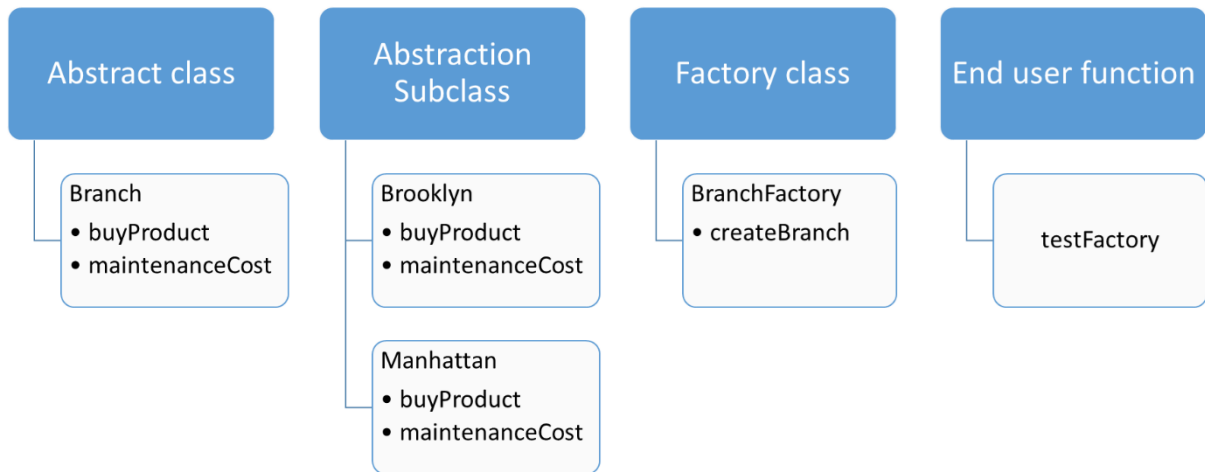




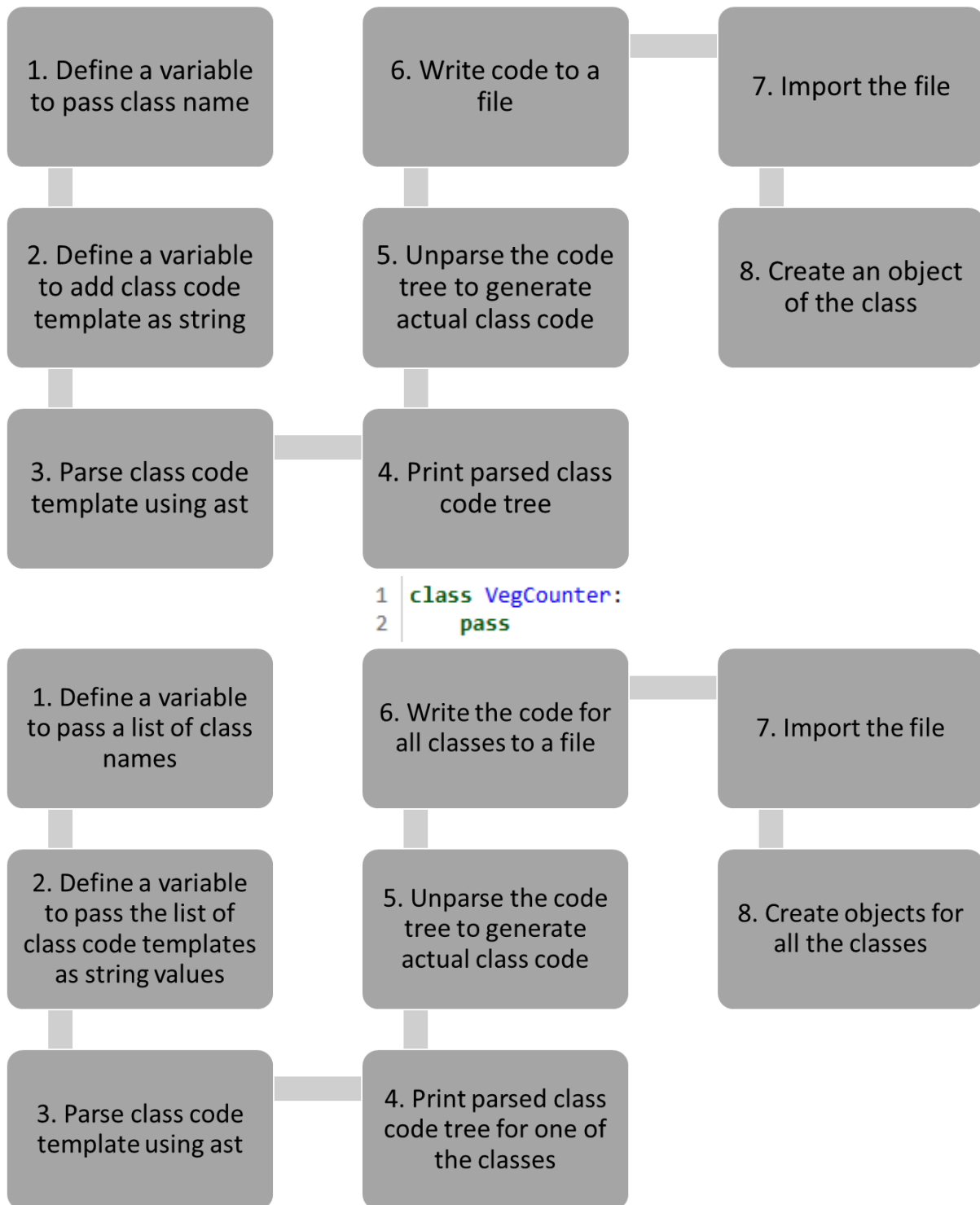
Chapter 13: Applying GOF Design Patterns – Part 2







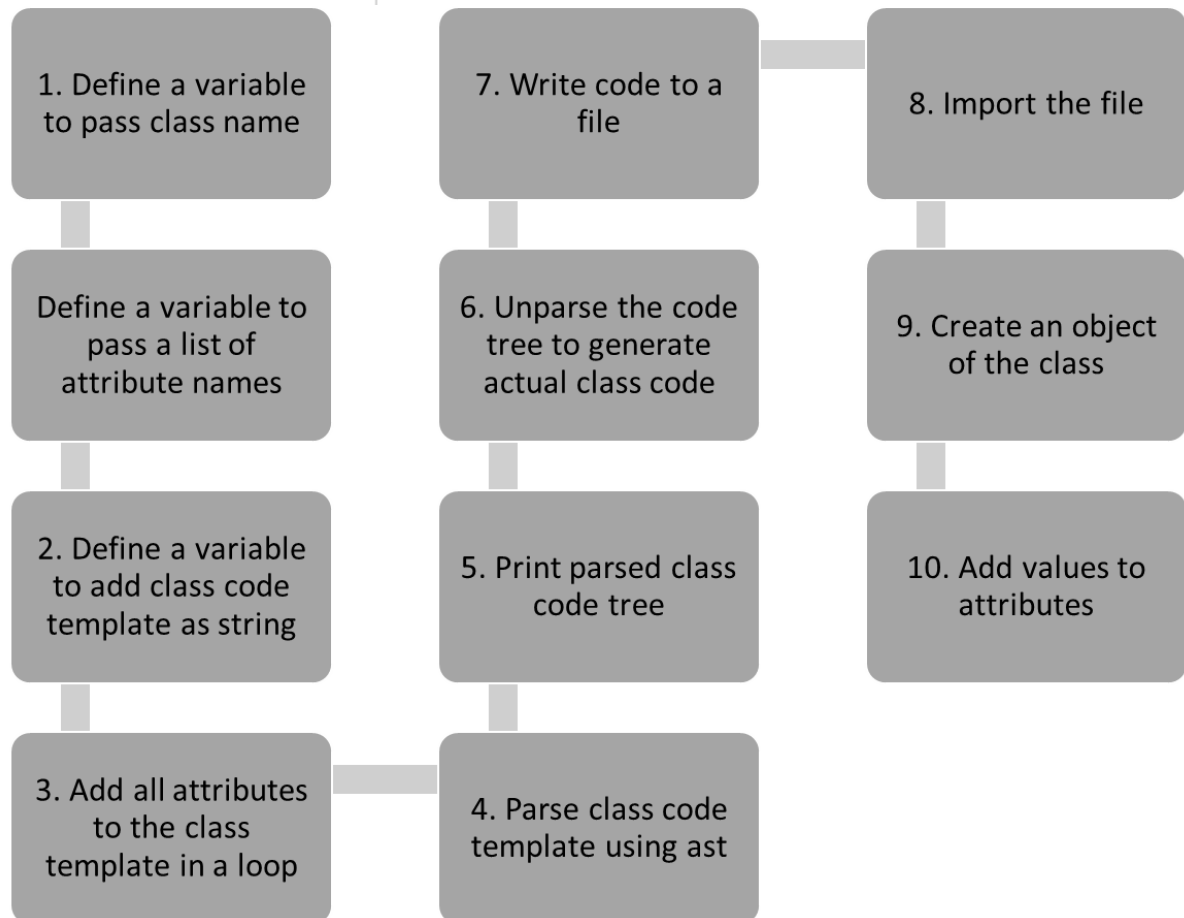
Chapter 14: Generating Code from AST



```

1  class VegCounter:
2      pass
3
4  class ElectronicsCounter:
5      pass
6
7  class PasadenaBranch:
8      pass
9
10 class VegasBranch:
11     pass

```



```

body=[
    FunctionDef(
        name='__init__',
        args=arguments(
            posonlyargs=[],
            args=[
                arg(arg='self')],
            vararg=arg(arg='items'),
            kwonlyargs=[],
            kw_defaults=[],
            defaults=[]),
        body=[
            FunctionDef(
                name='returnCart',
                args=arguments(
                    posonlyargs=[],
                    args=[
                        arg(arg='self')],
                    vararg=arg(arg='items'),
                    kwonlyargs=[],
                    kw_defaults=[],
                    defaults=[]),
                body=[

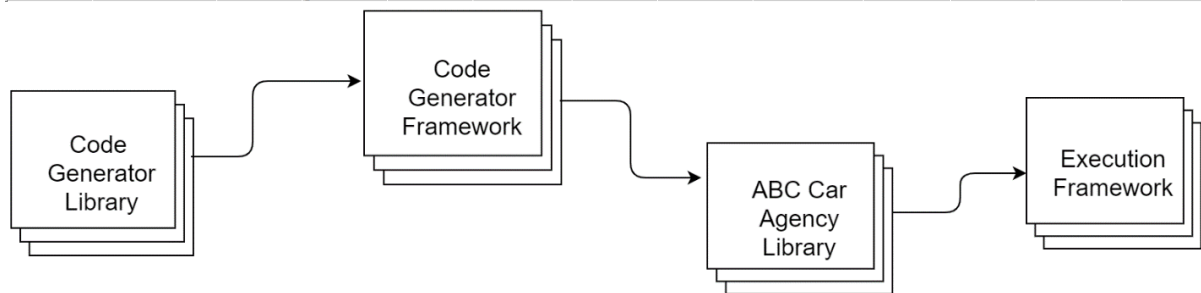
```

1
2
3
4

class VegCounter:
 items = None
 countername = None
 billamount = None

Chapter 15: Implementing a Case Study

symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height
3 ?		alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8
3 ?		alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8
1 ?		alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4
2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3
2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3
2 ?		audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1
1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7
1 ?		audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7
1	158	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9
0 ?		audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52
2	192	bmw	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3
0	192	bmw	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3
0	188	bmw	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3
0	188	bmw	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3
1 ?		bmw	gas	std	four	sedan	rwd	front	103.5	189	66.9	55.7
0 ?		bmw	gas	std	four	sedan	rwd	front	103.5	189	66.9	55.7
0 ?		bmw	gas	std	two	sedan	rwd	front	103.5	193.8	67.9	53.7
0 ?		bmw	gas	std	four	sedan	rwd	front	110	197	70.9	56.3
2	121	chevrolet	gas	std	two	hatchback	fwd	front	88.4	141.1	60.3	53.2
1	98	chevrolet	gas	std	two	hatchback	fwd	front	94.5	155.9	63.6	52
0		81 chevrolet	gas	std	four	sedan	fwd	front	94.5	158.8	63.6	52



```

classDiagram
    class CarSpecs {
        attributes...
        methods...
    }
    class CarMake {
        attributes...
        methods...
    }
    class CarCatalogue {
        attributes...
        methods...
    }
    class BodyStyle {
        attributes...
        methods...
    }
    class SaleType {
        attributes...
        methods...
    }
    class Chevrolet {
        attributes...
        methods...
    }
    class Mazda {
        attributes...
        methods...
    }
    class Bmw {
        attributes...
        methods...
    }
    class More... {
        attributes...
        methods...
    }
    class Wagon {
        attributes...
        methods...
    }
    class Sedan {
        attributes...
        methods...
    }
    class HatchBack {
        attributes...
        methods...
    }
    class HardTop {
        attributes...
        methods...
    }
    class Convertible {
        attributes...
        methods...
    }
    class New {
        attributes...
        methods...
    }
    class Resale {
        attributes...
        methods...
    }
    CarSpecs "1" -- "*" CarMake
    CarSpecs "1" -- "*" CarCatalogue
    CarSpecs "1" -- "*" BodyStyle
    CarSpecs "1" -- "*" SaleType
    CarSpecs "1" -- "*" Chevrolet
    CarSpecs "1" -- "*" Mazda
    CarSpecs "1" -- "*" Bmw
    CarSpecs "1" -- "*" More...
    CarSpecs "1" -- "*" Wagon
    CarSpecs "1" -- "*" Sedan
    CarSpecs "1" -- "*" HatchBack
    CarSpecs "1" -- "*" HardTop
    CarSpecs "1" -- "*" Convertible
    CarSpecs "1" -- "*" New
    CarSpecs "1" -- "*" Resale
  
```

The diagram illustrates the structure of the carsales system. At the top is the **CarSpecs** class, which is associated with several other classes: **CarMake**, **CarCatalogue**, **BodyStyle**, **SaleType**, **Chevrolet**, **Mazda**, **Bmw**, **More...**, **Wagon**, **Sedan**, **HatchBack**, **HardTop**, **Convertible**, **New**, and **Resale**. Each class has its own set of attributes and methods, represented by the boxes below the class name. The relationships are indicated by lines connecting the CarSpecs class to each of the other classes.

	classname	fuel_type	aspiration	num_of_door	drive_wheels	wheel_base	length	width	height	curb_weight
1	AlfaRomero1	gas	std	two	rwd	94.5	171.2	65.5	52.4	2823
24	Mitsubishi24	gas	std	two	fwd	93.7	157.3	64.4	50.8	1918
26	Nissan26	gas	std	two	fwd	94.5	165.3	63.8	54.5	1889
21	MercedesBenz21	diesel	turbo	two	rwd	106.7	187.5	70.3	54.9	3495
9	Dodge9	gas	std	four	fwd	93.7	157.3	63.8	50.6	1989

	classname	engine_location	engine_type	num_of_cylinders	engine_size	bore	stroke	compression_ratio	horse_power
13	Wagon13	front	ohc	four	92	2.92	3.41	9.2	76
9	Sedan9	front	ohc	four	90	2.97	3.23	9.4	68
16	Sedan16	front	dohc	six	258	3.63	4.17	8.1	176
5	Sedan5	front	ohc	four	108	3.5	2.8	8.8	101
18	Sedan18	front	ohc	four	91	3.03	3.15	9.0	68

```

79 class AlfaRomero0(CarMake, CarCatalogue, metaclass=CarSpecs):
80     fuel_type = 'gas'
81     aspiration = 'std'
82     num_of_door = 'two'
83     drive_wheels = 'rwd'
84     wheel_base = '88.6'
85     length = '168.8'
86     width = '64.1'
87     height = '48.8'
88     curb_weight = '2548'
89     fuel_system = 'mpfi'
90     city_mpg = '21'
91     highway_mpg = '27'
92     price = '13495'
93
94     def define_color(self):
95         BOLD = '\x1b[5m'
96         BLUE = '\x1b[94m'
97         return BOLD + BLUE
98
99     def define_spec(self):
100         specs = [self.fuel_type, self.aspiration, self.num_of_door, self.drive_wheels, self.wheel_base, self.length, self.width,
101                 self.height, self.curb_weight, self.fuel_system, self.city_mpg, self.highway_mpg]
102         return specs
103
104     def print_catalogue(self):
105         for i in self.define_spec():
106             print(self.define_color() + i['feature'], ': ', self.define_color() + i['info'])

```

Chapter 16: Following Best Practices

```
cart = greater.return_cart('paper clips','blue pens','stapler','pencils','a4paper','a3paper','chart','  
cart = greater.return_cart('paper clips','blue pens','stapler','pencils','a4paper','a3paper',  
'chart','sketch pens','canvas','water color','acrylic colors')  
cart = greater.return_cart('paper clips','blue pens','stapler','pencils','a4paper','a3paper',  
                             'chart','sketch pens','canvas','water color','acrylic colors')
```

```
1 def return_cart(*items):  
2     '''  
3     This function returns the list of items added to the cart.  
4     items: input the cart items. Eg: 'pens', 'pencils'  
5     '''  
6     cart_items = []  
7     for i in items:  
8         cart_items.append(i)  
9     return cart_items
```

```
import vegcounter as vc
```

```
vc.return_cart()
```

Signature: `vc.return_cart(*items)`

Docstring:

This function returns the list of items added to the cart.
items: input the cart items. Eg: 'pens', 'pencils'

BranchMetaclass

Init signature: `BranchMetaclass(classname, baseclasses, attributes)`

Docstring:

This is a meta class for ABC Megamart branch that adds an additional
quality to the attributes of branch classes.

Add this as only a meta class.

There are no methods to inherit this class as a parent class or super class.