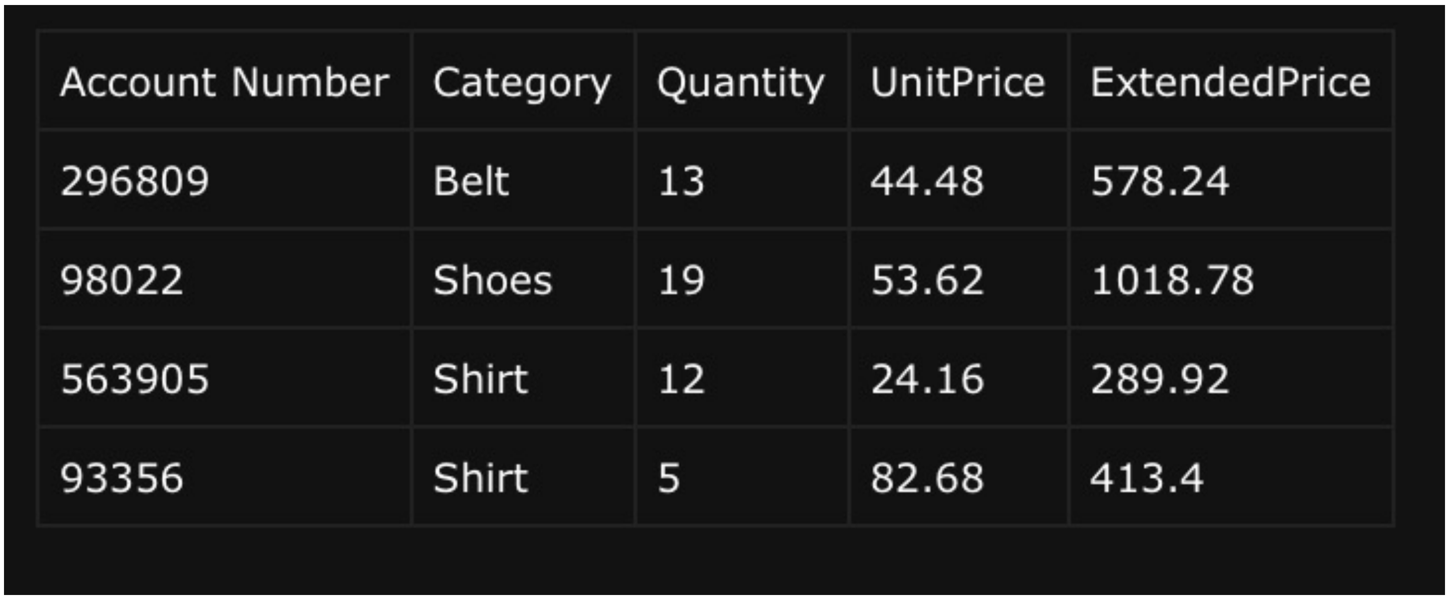
```
class Policy:
    def __init__(self,pnum,date,pname,paddr,premiumAmount):
        self._PolicyNumber = pnum
        self._PolicyDate = date
        self._PolicyName = pname
        self._PolicyAddr = paddr
        self._PolicyPremium = premiumAmount
    def getNumber(self):
        return self._PolicyNumber
    def getName(self):
        return self._PolicyName
    def getAddr(self):
        return self._PolicyAddr
    def getPremium(self):
        return self._PolicyPremium
    def getDate(self):
        return self._PolicyDate
    def setNumber(self,num):
        self._PolicyNumber = num
    def setName(self,name):
        self._PolicyName = name
    def setAddr(self,addr):
        self._PolicyAddr = addr
    def setPremium(self,premium):
        self._PolicyPremium = premium
    def setDate(self,d):
        self._PolicyDate = d
    def __str__(self):
        astring = "Policy Number:  " + self._PolicyNumber
        astring += "\nPolicy Name:  " + self._PolicyName
        astring += "\nPolicy Address:  " + self._PolicyAddr
        astring += "\nPolicy Premiumn:  " + str(self._PolicyPremium)
        return astring
if __name__== "__main__":
    aPolicy = Policy("23187","03-04-2019","John Jones","3 River St. N.Y., N.Y", 100)
    print (aPolicy)
    aPolicy.setNumber("3333")
    aPolicy.setDate("12-25-2018")
    aPolicy.setAddr("5 Highway 1")
    aPolicy.setName("Joe Thomas")
    aPolicy.setPremium(99)
    print (aPolicy.getNumber(),aPolicy.getDate(),aPolicy.getAddr(),aPolicy.getPremium())
```

```
#CompanySales


from SalesItem import SalesItem

class CompanySales:
    def __init__(self,companyname="",companyaddress="",filename=""):
        self.CompanyName=companyname
        self.CompanyAddress=companyaddress
        self.SalesList=[]
        if filename!="":
         try:
             file=open(filename,'r')
         except IOError:
             print ("File name is invalid")
             sys.exit()

         line = file.readline() #skip heading line
         for line in file:
            alist=line.split(',')
            salesObject = SalesItem(alist[0],alist[1],int(alist[2]),float(alist[3]),float(alist[4]))
            self.SalesList.append(salesObject)

    def setCompanyName(self,nname):
        self.CompanyName=nname
    def setCompanyAddress(self,naddy):
        self.CompanyAddress=naddy
    def getCompanyName(self):
        return self.CompanyName
    def getCompanyAddress(self):
        return self.CompanyAddress

    def __str__(self):
        return 'Company Name: '+str(self.CompanyName)+'\nCompany Address: '+str(self.CompanyAddress)

    def filterData(self,cat):

        filterdatalist=[]
        for object in self.SalesList:
            if object.getCategory()==cat:
                filterdatalist.append(object)
        return filterdatalist
    def Stats(self,thelist):
        maximum=0.0
        minimum=100000.0
        total=0.0

        for x in range(len(thelist)):
            price = thelist[x].getExtendedPrice()
            if price >maximum:
                maximum = price
            if price < minimum:
                minimum = price
            total+= price
        print('Max: ',maximum,'\nMin: ',minimum,'\nSum: ',round(total,2))

    def deepCopy(self,anotherObject):
        self.CompanyAddress = anotherObject.getCompanyAddress()
        self.CompanyName = anotherObject.getCompanyName()
        for i in range (len(anotherObject.SalesList)):
            if price >maximum:
                maximum = price
            if price < minimum:
                minimum = price
            total+= price
        print('Max: ',maximum,'\nMin: ',minimum,'\nSum: ',round(total,2))

    def deepCopy(self,anotherObject):
        self.CompanyAddress = anotherObject.getCompanyAddress()
        self.CompanyName = anotherObject.getCompanyName()
        for i in range (len(anotherObject.SalesList)):
            acctnum = anotherObject.SalesList[i].getAccountNumber()
            cat = anotherObject.SalesList[i].getCategory()
            quantity = anotherObject.SalesList[i].getQuantity()
            unitp = anotherObject.SalesList[i].getUnitPrice()
            extp = anotherObject.SalesList[i].getExtendedPrice()
            newObject = SalesItem(acctnum,cat,quantity,unitp,extp)
            self.SalesList.append(newObject)


if __name__=='__main__':
    object1=CompanySales('Buyers Inc.', '1000 KingsHighway', 'CompanySalesData.csv')
    newlist = object1.filterData("Shirt")
    for item in newlist:
        print (item)
    object1.Stats(newlist)
    # start test of shallow copy and deep copy
    print("Object1:",object1)
    object2 = CompanySales()
    print ("Object2:",object2)
    object2 = object1 #shallow copy
    print ("Object2 extended price",object2.SalesList[0].getExtendedPrice())
    object1.SalesList[0].setExtendedPrice(2.00)
    print ("Object1 SalesList[0] extended price",object1.SalesList[0].getExtendedPrice())
    print ("Object2 SalesList[0] extended price",object2.SalesList[0].getExtendedPrice())
    object3 = CompanySales()
    object3.deepCopy(object1)
    print ("Object 1",object1)
    print ("Object 2",object3)
    object3.setCompanyName("Wallmart")
    print (object1)
    print (object3)
    print (object1.SalesList[1].getUnitPrice())
    object1.SalesList[1].setUnitPrice(111.00)
    #shows a deepcopy
    print (object1.SalesList[1].getUnitPrice())
    print (object3.SalesList[1].getUnitPrice())
```

```
#company sales

class SalesItem:
    def __init__(self,num='',category='',quantity=0,UnitPrice=0.0,ExtendedPrice=0.0):
        self.AccountNumber=num
        self.category=category
        self.quantity=int(quantity)
        self.UnitPrice=float(UnitPrice)
        self.ExtendedPrice=float(ExtendedPrice)
    def setAccountNumber(self,setnum):
        self.AccountNumber=setnum
    def setCategory(self,setcategory):
        self.category=setcategory
    def setQuantity(self,setquantity):
        self.quantity=setquantity
    def setUnitPrice(self,setunitprice):
        self.UnitPrice=setunitprice
    def setExtendedPrice(self,setextendedprice):
        self.ExtendedPrice=setextendedprice
    def getAccountNumber(self):
        return self.AccountNumber
    def getCategory(self):
        return self.category
    def getQuantity(self):
        return self.quantity
    def getUnitPrice(self):
        return self.UnitPrice
    def getExtendedPrice(self):
        return self.ExtendedPrice
    def __str__(self):
        return "Account Number "+ self.AccountNumber +'\n'+'Category '+ self.category+'\n'+'Quantity
'+str(self.quantity)+'\n'+ 'Unit Price '+str(self.UnitPrice)+'\n'+'Extended Price
'+str(self.ExtendedPrice)
```



Account Number	Category	Quantity	UnitPrice	ExtendedPrice
296809	Belt	13	44.48	578.24
98022	Shoes	19	53.62	1018.78
563905	Shirt	12	24.16	289.92
93356	Shirt	5	82.68	413.4