



CPE 231 Database Systems

Lab Assignment 2 for Lab Weeks 4-5.

Score: ____/30

Due Lab Week 6 (2020-Sep-21). Demo to Lab Instructor. Submit code on MyLE.
Computer Engineering Department, KMUTT.

This lab will be based on the following example data for 3 classes (objects) of Product, Customer, and Invoice:

Product Class:

| *Code | Name | Units |
|-------|--------------------------|-------|
| HD01 | Seagate HDD 80 GB | PCS |
| HD02 | IBM HDD 60 GB | PCS |
| INT01 | Intel Pentium IV 3.6 GHz | PCS |

Customer Class:

| *Customer Code | Name | Address | Credit Limit | Country |
|----------------|------------------|-----------------------|--------------|----------|
| Sam | Sam Co., Ltd. | 122 Bangkok | 500,000 | Thailand |
| CP | Charoen Pokaphan | 14 Sukhumvit, Bangkok | 2,000,000 | Thailand |

Invoice Class:

| | | | | | | | | | | | | | | | | | | |
|---|----------|---------------|----------------|----------|--------|------------|---------------|----------|------------|----------------|------|---|----------|----------|------|---|----------|----------|
| *Invoice No | Date | Customer Code | Due Date | Total | VAT | Amount Due | | | | | | | | | | | | |
| IN100/20 | 2/1/20 | Sam | | 8,000.00 | 560.00 | 8,560.00 | | | | | | | | | | | | |
| <table><tr><td>*Product Code</td><td>Quantity</td><td>Unit Price</td><td>Extended Price</td></tr><tr><td>HD01</td><td>2</td><td>3,000.00</td><td>6,000.00</td></tr><tr><td>HD02</td><td>1</td><td>2,000.00</td><td>2,000.00</td></tr></table> | | | | | | | *Product Code | Quantity | Unit Price | Extended Price | HD01 | 2 | 3,000.00 | 6,000.00 | HD02 | 1 | 2,000.00 | 2,000.00 |
| *Product Code | Quantity | Unit Price | Extended Price | | | | | | | | | | | | | | | |
| HD01 | 2 | 3,000.00 | 6,000.00 | | | | | | | | | | | | | | | |
| HD02 | 1 | 2,000.00 | 2,000.00 | | | | | | | | | | | | | | | |
| IN101/20 | 4/1/20 | Sam | | 2,000.00 | 140.00 | 2,140.00 | | | | | | | | | | | | |
| <table><tr><td>*Product Code</td><td>Quantity</td><td>Unit Price</td><td>Extended Price</td></tr><tr><td>HD02</td><td>1</td><td>2,000.00</td><td>2,000.00</td></tr></table> | | | | | | | *Product Code | Quantity | Unit Price | Extended Price | HD02 | 1 | 2,000.00 | 2,000.00 | | | | |
| *Product Code | Quantity | Unit Price | Extended Price | | | | | | | | | | | | | | | |
| HD02 | 1 | 2,000.00 | 2,000.00 | | | | | | | | | | | | | | | |

The lab instructor will provide you with a python program for you to use as an example to learn from. This program has the specifications and functionality described below to 1) define classes related to the invoice form and 2) define create, read, update, and delete (CRUD) API function calls to these classes. Your assignment is to extend the provided python program to include the Receipt Form data.

Product Class:

- Dictionary with key of Product Code that will have values as another dictionary with keys of Name and Units. Example of this dictionary for the above data:

```
{'HD01': {'Name': 'Seagate HDD 80 GB', 'Units': 'PCS'},
'HD02': {'Name': 'IBM HDD 60 GB', 'Units': 'PCS'},
'INT01': {'Name': 'Intel Pentium IV 3.6 GHz', 'Units': 'PCS'}}
```

Customer Class:

- Dictionary with key of Customer Code and have values as another dictionary with keys of Name, Address, Credit Limit, and Country.

Invoice Class:

- Dictionary with Key of Invoice No and values as another dictionary with keys of Date, Customer Code, Due Date, Total, VAT, Amount Due, and a list of line items where each item in the list is a dictionary of Product Code, Quantity, Unit Price, and Product Total.

```
{'IN100/20': { 'Date': '2020-01-02', 'Customer Code': 'Sam', 'Due Date': Null, 'Total': 8000.00,
'VAT': 560.00, 'Amount Due': 8560.00,
'Items List': [ {'Product Code': 'HD01', 'Quantity': 2, 'Unit Price': 3000.00,
'Extended Price': 6000.00}, {'Product Code': 'HD02', 'Quantity': 1,
'Unit Price': 2000.00, 'Extended Price': 2000.00} ]
},
```

```
'IN101/20': { 'Date': '2020-01-04', 'Customer Code': 'Sam', 'Due Date': Null, 'Total': 2000.00,
```

```

        'VAT': 140.00, 'Amount Due': 2140.00,
        'Items List': [{ 'Product Code': 'HD02', 'Quantity': 1, 'Unit Price': 2000.00,
                        'Extended Price': 2000.00 } ]
    }
}

```

Create the CRUD (Create, Read, Update, Delete) function calls (API) for the above 3 classes:

- `def create_product(products, code, name, units)`. Adds the new product record to products object (dictionary) using parameters code, name, units.
- `def read_product(products, code)`. Finds the product code in products object and returns 1 record in dictionary form of { 'Name': ___, 'Units': ___ }
- `def update_product (products, code, newName, newUnits)`. Finds the product code in products object and then changes the name and units to the values in newName, and newUnits.
- `def delete_product(products, code)`. Finds the product code in products object and removes it from the dictionary.
- `def create_customer(customers, customerCode, customerName, address, creditLimit, country)`. Adds the new customer record to customers object (dictionary).
- `def read_customer(customers, customerCode)`. Finds the customer code in customers object and returns 1 record in dictionary form.
- `def update_customer (customers, customerCode, newCustomerName, newAddress, newCreditLimit, newCountry)`. Finds the customer code in customers object and then changes the values to the new ones.
- `def delete_customer(customers, customerCode)`. Finds the customer code in customers object and removes it from the dictionary.
- `def create_invoice(invoices, invoiceNo, invoiceDate, customerCode, dueDate, invocieLineTuplesList)`. Adds the new invoice record to invoices object (dictionary). Note that the function will calculate Total, VAT, and Amount Due from the data in the invoiceLineDictList parameter. The invoiceLineDictList data will be a list of dictionary, where each dictionary item of the list is in this example format: { 'Product Code': 'HD01', 'Quantity': 2, 'Unit Price': 3000.00 }. Note that for each line item the Extended Price will be calculated by the function using Quantity * Unit Price.
- `def read_invoice(invoices, invoiceNo)`. Finds the invoice number in invoices object and returns 1 invoice record in dictionary form.
- `def update_invoice(invoices, invoiceNo, newInvoiceDate, newCustomerCode, newDueDate, newInvocieLineTuplesList)`. Finds the invoice number in invoices object and then changes the values to the new ones.
- `def delete_invoice(invoices, invoiceNo)`. Finds the invoice number in invoices object and removes it from the dictionary.
- `def update_invoice_line(invoices, invoiceNo, productCode, newQuantity, newUnitPrice)`. The line item of this invoice number is updated for this product code. Note that the extended price must also be recalculated, after which all the related data in the invoice must be updated such as Total, VAT, and Amount Due.
- `def delete_invoice_line(invoices, invoiceNo, productCode, newQuantity, newUnitPrice)`. The line item of this invoice number is updated to delete this product code. Note that all the related data in the invoice must be updated such as Total, VAT, and Amount Due.
- `def report_list_all_invoices(invoices, customers, products)`. Will dump all invoices data in easy to read form. Please show the customer name and product name also.
- `def report_list_all_products(products)`. Will dump all products data in easy to read form.
- `def report_list_all_customers(customers)`. Will dump all customers data in easy to read form.

- `def report_products_sold(invoices, products, dateStart, dateEnd)`. Will list in products sold in the given date range in tabular format of: Product Code, Product Name, Total Quantity Sold, Total Value Sold. Here, (product code) will be unique. At the end also show the sum of Total Value Sold.
- `def report_customer_products_sold_list(invoices, products, customers, dateStart, dateEnd)`. Will list customers and list the products sold to them in the given date range in this format: Customer Code, Customer Name, Product Code, Product Name, Invoice No, Invoice Date, Quantity Sold, Value Sold. Here, (customer code, product code, invoice no) will be unique. At the end also show the sum of Value Sold.
- `def report_customer_products_sold_total(invoices, products, customers, dateStart, dateEnd)`. Will list customers and the total number and value of products sold to them in the given date range in this format: Customer Code, Customer Name, Product Code, Product Name, Total Quantity Sold, Total Value Sold. Here (customer code, product code) will be unique. At the end also show the sum of Total Value Sold.

Your Assignment

Receipt Form

*Receipt No.: RCT1002/20 *Date: 05/02/2020

*Customer Code: Sam Customer Name: Sam Co., Ltd.

*Payment Method: Credit Card Payment Reference: Master Card, Citibank

Total Received: 10,000.00

Remarks: Partially paid on IN101/20

Received for these Sales Invoices:

| # | *Invoice No. | Invoice Date | Invoice Full Amount | Invoice Amount Remain | *Amount Paid Here |
|-----------------|--------------|--------------|---------------------|-----------------------|-------------------|
| 1 | IN100/20 | 02/01/2020 | 8,560.00 | 8,560.00 | 8,560.00 |
| 2 | IN101/20 | 04/01/2020 | 2,140.00 | 2,140.00 | 1,440.00 |
| 3 | | | | | |
| 4 | | | | | |
| Total Received: | | | | | 10,000.00 |

Add the Receipt and Payment Method classes to your program. Write the following additional API functions:

- 4 CRUD functions for Payment Method class.
- 4 CRUD functions for Receipt class plus an Update and a Delete function for the receipt line items.
- `def report_list_all_receipts(receipts, invoices, customers)`. Please show customer name in the header and also the invoice date as part of invoice information in line item. **invoice Header**
- `def report_unpaid_invoices(invoices, customers, receipts)`. Shows a list of invoices with amount remaining with these fields: Invoice Number, Invoice Date, Customer Name, Invoice Amount Due, Invoice Amount Received, Invoice Amount Not Paid. At the end also show the total of Invoice Amount Not Paid. The receipts object will be used to calculate Invoice Amount Received for each invoice.

Submission

Show run during class to lab instructors on the 6th Week of Lab. You will use python scripts to run your API calls to show that your program correctly works in creating, changing, and reporting each class data. Then submit your source code including sample run with scripts used and output created as a PDF file via upload to MyLE under topic "Lab Week 2" before 2020-Sep-21 at time 23:59. Your score will be based on the run demo, but the PDF file will be used to check for copied work.

API → 169 def report list

1 2 3 4 CRUD functions for Payment Method class.

Code หน้า lab2_receipt_app.py + แสดงผลในโค้ด

```

84 #ข้อ 1
85 #Create_paymentMethod
86
87 print("***50,Create Payment Method","**50)
88 paymentMethod = PaymentMethod()
89 create_payment_method(paymentMethod, "4", "e-Banking")
90 read_payment_method(paymentMethod,"4")
91 print("")
92
93
94 #update_paymentMethod
95 print("***50,Update Payment Method","**50)
96 update_payment_method(paymentMethod, "4", "Banking")
97 read_payment_method(paymentMethod,"4")
98 print("")
99
100 #delete_paymentMethod
101 print("***50,delete Payment Method ","**50)
102 delete_payment_method(paymentMethod, "4")
103 report_list_payment_method(paymentMethod)
104 print("")
105
106 #Create_paymentMethod
107 print("***50,Create Payment Method","**50)
108 paymentMethod = PaymentMethod()
109 create_payment_method(paymentMethod, "4", "e-Banking")
110 read_payment_method(paymentMethod,"4")
111 print("")
112

```

```

***** Create Payment Method *****
Payment Method Success.
{'Name': 'e-Banking'}
***** Update Payment Method *****
Payment Method Success.
{'Name': 'Banking'}
***** delete Payment Method *****
Product Delete Success.
{}
***** Create Payment Method *****
Payment Method Success.
{'Name': 'e-Banking'}

```


- 4 CRUD functions for Receipt class plus an Update and a Delete function for the receipt line items.

code หน้า Receipt.py

```

New folder > Receipt.py
20 ① def create(self, receiptNo, receiptDate, customerCode, paymentMethod, paymentReference, remark, receiptLineItemList):
21     if receiptNo in self.dict:
22         return {'Is Error': True, 'Error Message': "Receipt No '{}' already exists. Cannot Create.".format(receiptNo)}
23     else:
24         receiptLineItemList, total = self.__updateLineItem(receiptLineItemList)
25         self.dict[receiptNo] = {"Receipt Date": receiptDate, "Customer Code": customerCode, "Payment Method": paymentMethod, "Payment Reference": paymentReference,
26         return {'Is Error': False, 'Error Message': ""}
27
28
29 ② def read(self, receiptNo):
30     if receiptNo in self.dict:
31         retreceipt = self.dict[receiptNo]
32     else:
33         return ({'Is Error': True, 'Error Message': "Receipt No '{}' not found. Cannot Read.".format(receiptNo)}, {})
34
35     return ({'Is Error': False, 'Error Message': ""}, retreceipt)
36
37 ③ def update(self, receiptNo, newReceiptDate, newCustomerCode, newPaymentMethod, newPaymentReference, newRemark, newReceiptLineItemList):
38     if receiptNo in self.dict:
39         self.dict[receiptNo]["Receipt Date"] = newReceiptDate
40         self.dict[receiptNo]["Customer Code"] = newCustomerCode
41         self.dict[receiptNo]["Payment Method"] = newPaymentMethod
42         self.dict[receiptNo]["Payment Reference"] = newPaymentReference
43         self.dict[receiptNo]["Remark"] = newRemark
44
45         self.dict[receiptNo]["Items List"] = newReceiptLineItemList
46     else:
47         return {'Is Error': True, 'Error Message': "Receipt No '{}' not found. Cannot Update.".format(receiptNo)}
48
49     return {'Is Error': False, 'Error Message': ""}
50
51 ④ def delete(self, receiptNo):
52     if receiptNo in self.dict:
53         del self.dict[receiptNo]
54     else:
55         return {'Is Error': True, 'Error Message': "Receipt No '{}' not found. Cannot Delete.".format(receiptNo)}
56     return {'Is Error': False, 'Error Message': ""}
57
58 New folder > Receipt.py
59
60 ⑤ def update_receipt_line(self, receiptNo, invoiceNo, amountPaid):
61     if receiptNo in self.dict:
62         receiptLineItemList = []
63         bUpdated = False
64         for lineItem in self.dict[receiptNo]["Items List"]:
65             invoiceLineItem = {}
66             if lineItem["Invoice No"] == invoiceNo:
67                 invoiceLineItem["Invoice No"] = invoiceNo
68                 invoiceLineItem["Amount Paid Here"] = amountPaid
69
70                 receiptLineItemList.append(invoiceLineItem)
71                 bUpdated = True
72             else:
73                 receiptLineItemList.append(lineItem)
74         print(receiptLineItemList)
75         if bUpdated:
76             receiptLineItemList, total = self.__updateLineItem(receiptLineItemList)
77             self.dict[receiptNo]["Items List"] = receiptLineItemList
78             self.dict[receiptNo]["Total Received"] = total
79         else:
80             return {'Is Error': True, 'Error Message': "Receipt Code '{}' not found in Invoice No '{}'. Cannot Update.".format(receiptNo, invoiceNo)}
81     else:
82         return {'Is Error': True, 'Error Message': "Receipt No '{}' not found. Cannot Update.".format(receiptNo)}
83
84     return {'Is Error': False, 'Error Message': ""}
85
86
87 ⑥ def delete_receipt_line(self, receiptNo, invoiceNo):
88     # The line item of this invoice number is updated to delete this product code.
89     # Note that all the related data in the invoice must be updated such as Total, VAT, and Amount Due.
90     # Returns dictionary {'Is Error': ___, 'Error Message': ___}
91     if receiptNo in self.dict:
92         total = 0
93         receiptLineItemList = []
94         bDeleted = False
95         for lineItem in self.dict[receiptNo]["Items List"]:
96             if lineItem["Invoice No"] == invoiceNo:
97                 bDeleted = True
98
99
100         if bDeleted:
101             receiptLineItemList, total = self.__updateLineItem(receiptLineItemList)
102             self.dict[receiptNo]["Items List"] = receiptLineItemList
103             self.dict[receiptNo]["Total Received"] = total
104
105         else:
106             return {'Is Error': True, 'Error Message': "Receipt Code '{}' not found in Invoice No '{}'. Cannot Delete.".format(receiptNo, invoiceNo)}
107     else:
108         return {'Is Error': True, 'Error Message': "Receipt No '{}' not found. Cannot Delete.".format(receiptNo)}
109
110     return {'Is Error': False, 'Error Message': ""}
111

```

```

New folder > lab2_invoice_app.py
113 ① #Create_receipt
114     print("***50,\"Create\", \"***50)
115     receipt = Receipt()
116     create_receipt(receipt, "RCT1002/20",
117                   '2020-02-05',
118                   "Sam", "Credit Card",
119                   "Master Card, Citibank",
120                   "Partially paid on IN101/20", [{"Invoice No" : 'IN100/20',"Amount Paid Here" : 8560}, {"Invoice No" : 'IN101/20',"Amount Paid Here" : 1440}
121     read_receipt(receipt, "RCT1002/20")
122     print("")
123
124 ② #update_receipt
125     print("***50,\"Update\", \"***50)
126     update_receipt(receipt, "RCT1002/20", "2020-02-06", "Sam", "Credit Card",
127                   "Master Card, Citibank",
128                   "Partially paid on IN101/20", [{"Invoice No" : 'IN100/20',"Amount Paid Here" : 8560}, {"Invoice No" : 'IN101/20',"Amount Paid Here" : 1440}
129     read_receipt(receipt, "RCT1002/20")
130     print("")
131
132 ③ #update_receipt_line
133     print("***50,\"update_receipt_line \", \"***50)
134     update_receipt_line(receipt, "RCT1002/20", "IN100/20", 8000)
135     read_receipt(receipt, "RCT1002/20")
136     print("")
137
138 ④ #delete_receipt_line
139     print("***50,\"delete_receipt_line \", \"***50)
140     delete_receipt_line(receipt, "RCT1002/20", "IN100/20")
141     read_receipt(receipt, "RCT1002/20")
142     print("")
143
144 ⑤ #delete_receipt
145     print("***50,\"delete_receipt \", \"***50)
146     delete_receipt(receipt, "RCT1002/20")
147     report_list_all_receipt(receipt, invoices, customers)
148     print("")
149
150 ⑥ -> สร้างตารางอีกรอบ เพื่อจะใส่เงิน ได้ดี ต่อปัจจัย กับ ข้อ 4 ได้
151     #Create_receipt
152     print("***50,\"Create receipt\", \"***50)
153     receipt = Receipt()
154     create_receipt(receipt, "RCT1002/20",
155                   '2020-02-05',
156                   "Sam", "Credit Card",
157                   "Master Card, Citibank",
158                   "Partially paid on IN101/20", [{"Invoice No" : 'IN100/20',"Amount Paid Here" : 8560}, {"Invoice No" : 'IN101/20',"Amount Paid Here" : 1440}
159     read_receipt(receipt, "RCT1002/20")
160     print("")

```

Run โค้ด แสดงผล

```

***** Create ***** ①
Receipt Create Success.
{'Receipt Date': '2020-02-05', 'Customer Code': 'Sam', 'Payment Method': 'Credit Card', 'Payment Reference': 'Master Card, Citibank', 'Total Received': 10000, 'Remarks': 'Partially paid on IN101/20', 'Items List': [{'Invoice No': 'IN100/20', 'Amount Paid Here': 8560}, {'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}]}

***** Update ***** ②
Receipt Update Success.
{'Receipt Date': '2020-02-06', 'Customer Code': 'Sam', 'Payment Method': 'Credit Card', 'Payment Reference': 'Master Card, Citibank', 'Total Received': 10000, 'Remarks': 'Partially paid on IN101/20', 'Items List': [{'Invoice No': 'IN100/20', 'Amount Paid Here': 8560}, {'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}], 'Remark': 'Partially paid on IN101/20'}

***** update_receipt_line ***** ③
[{'Invoice No': 'IN100/20', 'Amount Paid Here': 8000}, {'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}]
Invoice Line Item Update Success.
{'Receipt Date': '2020-02-06', 'Customer Code': 'Sam', 'Payment Method': 'Credit Card', 'Payment Reference': 'Master Card, Citibank', 'Total Received': 9440, 'Remarks': 'Partially paid on IN101/20', 'Items List': [{'Invoice No': 'IN100/20', 'Amount Paid Here': 8000}, {'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}], 'Remark': 'Partially paid on IN101/20'}

***** delete_receipt_line ***** ④
Receipt Line Item Delete Success.
{'Receipt Date': '2020-02-06', 'Customer Code': 'Sam', 'Payment Method': 'Credit Card', 'Payment Reference': 'Master Card, Citibank', 'Total Received': 1440, 'Remarks': 'Partially paid on IN101/20', 'Items List': [{'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}], 'Remark': 'Partially paid on IN101/20'}

***** delete_receipt ***** ⑤
Invoice Delete Success.

***** Create receipt ***** ⑥
Receipt Create Success.
{'Receipt Date': '2020-02-05', 'Customer Code': 'Sam', 'Payment Method': 'Credit Card', 'Payment Reference': 'Master Card, Citibank', 'Total Received': 10000, 'Remarks': 'Partially paid on IN101/20', 'Items List': [{'Invoice No': 'IN100/20', 'Amount Paid Here': 8560}, {'Invoice No': 'IN101/20', 'Amount Paid Here': 1440}]}

```


- def report_list_all_receipts(receipts, invoices, customers). Please show customer name in the header and also the invoice date as part of invoice information in line item.

หน้า code API.py

```

399 def report_list_all_receipt(receipt, invoices, customers):
400     allreceipt = receipt.dump()
401     # re-format result by add Customer Name in object and Product Name in line item
402     result = {}
403     for receiptNo, receiptDetail in allreceipt.items():
404         newValue = {}
405         for receiptColume, receiptData in receiptDetail.items():
406             if receiptColume == "Customer Code":
407                 newValue[receiptColume] = receiptData
408                 customer = customers.read(receiptDetail["Customer Code"])
409                 if customer[0]['Is Error']:
410                     newValue["Customer Name"] = ""
411                 else:
412                     newValue["Customer Name"] = customer[1]["Name"]
413
414             elif receiptColume == "Items List":
415                 newLineItemList = []
416                 for lineItem in receiptDetail['Items List']:
417                     newLineItem = {}
418                     for lineItemColume, lineItemData in lineItem.items():
419                         if lineItemColume == "Invoice No":
420                             invoice = invoices.read(lineItem["Invoice No"]) #join
421                             if invoice[0]['Is Error']:
422                                 invoice[1] = ""
423                             else:
424                                 invoice[1]["Invoice No"] = lineItem["Invoice No"]
425                                 newLineItem = invoice[1]
426                             newLineItemList.append(newLineItem)
427                     newValue[receiptColume] = newLineItemList
428             else:
429                 newValue[receiptColume] = receiptData
430         result[receiptNo] = newValue
431     print(result)
432     print(" ")
433     printForm(result) |
434     return result #send result for caller program to use
435

```

หน้า helper_functions.py

```

95 def printForm(dict): ฟังก์ชันในการปรับ รูปแบบ ตาราง
96     for i in dict:
97         print("Customer Name :",dict[i]["Customer Name"])
98         print(" ")
99         print("Invoice No",end = " , ")
100         print("Date",end = " , ")
101         print("Customer Code",end = " , ")
102         print("Due Date",end = " , ")
103         print("Total",end = " , ")
104         print("VAT",end = " , ")
105         print("Amount Due",end = " , ")
106         print("Items List")
107         print(" ")
108         print(" ")
109         items = dict[i]["Items List"]
110         for j in items:
111             print(j ["Invoice No"],end = " , ")
112             print(j ["Date"],end = " , ")
113             print(j ["Customer Code"],end = " , ")
114             print(j ["Due Date"],end = " , ")
115             print(j ["Total"],end = " , ")
116             print(j ["VAT"],end = " , ")
117             print(j ["Amount Due"],end = " , ")
118             print(j ["Items List"])
119
120     print(" ")
121     print(" ")
122     print(" ")
123

```

Code หน้า lab2_receipt_app.py + แสดงผลในโค้ด

```

126 report_list_all_receipt(receipt, invoices, customers)
127
128 #ข้อ 3
129 print("""50,3 is report unpaid invoices""")
130 report_unpaid_invoices(invoices, customers, receipt)
131 print("")
132

```

***** 3 is report unpaid invoices *****

| Invoice Number | Invoice Date | Customer Name | Invoice Amount Due | Invoice Amount Received | Invoice Amount Not Paid |
|------------------------------|--------------|---------------|--------------------|-------------------------|-------------------------|
| IN100/20 | 2020-01-02 | Sam Co., Ltd. | 8560.0 | 8560 | 0.0 |
| IN101/20 | 2020-01-04 | Sam Co., Ltd. | 2140.0 | 1440 | 700.0 |
| Total Unpaid invoice = 700.0 | | | | | |

- def report_unpaid_invoices(invoices, customers, receipts). Shows a list of invoices with amount remaining with these fields: Invoice Number, Invoice Date, Customer Name, Invoice Amount Due, Invoice Amount Received, Invoice Amount Not Paid. At the end also show the total of Invoice Amount Not Paid. The receipts object will be used to calculate Invoice Amount Received for each invoice.

Code หน้า API.PY

```

313 def report_unpaid_invoices(invoices, customers, receipts):
314     total = 0
315     allreceipt = receipts.dump()
316     receiptlist = []
317     for receiptNo, receiptDetail in allreceipt.items(): # Loop receipt
318         for receiptColumn in receiptDetail:
319             if receiptColumn == "Items List":
320                 for lineItem in receiptDetail['Items List']:
321                     invoiceDict = {}
322                     invoiceDict['Invoice No'] = lineItem['Invoice No']
323                     invoiceDict['Amount Paid Here'] = lineItem['Amount Paid Here']
324                     invoice = invoices.read(lineItem['Invoice No'])
325                     if invoice[0]['Is Error']:
326                         invoiceDict = {}
327                     else:
328                         customer = customers.read(invoice[1]['Customer Code'])
329                         if customer[0]['Is Error']:
330                             invoiceDict['Customer Name'] = ''
331                         else:
332                             invoiceDict['Customer Name'] = customer[1]['Name']
333                             invoiceDict['Date'] = invoice[1]['Date']
334                             invoiceDict['Amount Due'] = invoice[1]['Amount Due']
335
336                             dif = invoiceDict['Amount Due'] - invoiceDict['Amount Paid Here']
337                             total += dif
338                             invoiceDict['Invoice Amount Not paid'] = dif
339
340                 receiptlist.append(invoiceDict)
341     printUnPaid(receiptlist)
342     print("Total Unpaid invoice = ",total)
343     print(" ")
344     print("-----")
345

```

Code หน้า helper_functions.py

```

125
126 def printUnPaid(dict):
127     print(" ")
128     print("Invoice Number",end = " , ")
129     print("Invoice Date",end = " , ")
130     print("Customer Name",end = " , ")
131     print("Invoice Amount Due",end = " , ")
132     print("Invoice Amount Received",end = " , ")
133     print("Invoice Amount Not Paid")
134
135     print(" ")
136     print(" ")
137     for i in dict:
138         print(i ["Invoice No"],end = " , ")
139         print(i ["Date"],end = " , ")
140         print(i ["Customer Name"],end = " , ")
141         print(i ["Amount Due"],end = " , ")
142         print(i ["Amount Paid Here"],end = " , ")
143         print(i ["Invoice Amount Not paid"])
144     print(" ")
145

```

Code หน้า lab2_receipt_app.py + แสดงผลขึ้นโค้ด

```

167 #ข้อ 4
168 print("***50,4 is report unpaid invoices","**50)
169 report_unpaid_invoices(invoices, customers, receipt)
170 print("")
171
172
173
174 waitKeyPress("Above are results for creating 2 invoices and line item.")
175
176 except: #this traps for unexpected system errors
177     print ("Unexpected error:", sys.exc_info()[0])
178     raise # this line can be erased. It is here to raise another error so you can see which line to debug.
179 else:
180     print("Normal Termination. Goodbye!")
181

```

```

***** 4 is report unpaid invoices *****
Invoice Number , Invoice Date , Customer Name , Invoice Amount Due , Invoice Amount Received , Invoice Amount Not Paid
IN100/20 , 2020-01-02 , Sam Co., Ltd. , 8560.0 , 8560 , 0.0
IN101/20 , 2020-01-04 , Sam Co., Ltd. , 2140.0 , 1440 , 700.0
Total Unpaid invoice = 700.0

```