Instructions

* Create separate file for each programs
* Write appropriate comments for each program in methods, classes
* Write appropriate variable names, method names

Program - 1

* Write a program that contains ‘manufacturing’ class which allows product manufacturing.
* It purchases raw material from the market and produces a product.
* One should manage the raw material product to produce the actual product.
* One should manage the ratio of raw material to produce the actual product.
  + Example - 2 wheels(raw material) are required to produce 1 bicycle(actual product)
* Take input from user raw material, raw material ratio qty, actual product
* While producing the actual product, if the system doesn’t have enough of its stock of raw material, it should show a warning message as “Not enough raw material available to produce the product, please do the purchase”.
* Program should be menu driven allowing to
  + Purchase Raw Material Product
  + Manufacture Actual Product
  + Show Raw Material Quantity
  + Show Actual Product Quantity
  + Exit
* Following attributes and method function should be part of the class
  + Appropriate constructor to set the values for raw material product, actual product, raw material ratio qty value, raw material qty
  + produce() - taking no qty to be produced for actual product as argument
  + display\_raw\_material\_stock()
  + display\_final\_product\_stock()
  + purchase\_raw\_material() - taking no of qty of raw material to be purchased

Program - 2

* Write a program which extends Program - 1
* Add appropriate class and perform inheritance
* Add the facility of scrapping the raw material product and actual product
* Add appropriate option in the menu for
  + scrapping the raw material product
  + scrapping the actual product
* Add appropriate methods to scrap the raw material product and actual product

Program - 3

* Write a program which is having a class InventoryManagement.
* System should manage the quantity of purchase and sales processes.
* If the system is not having enough quantity during sales, the system should show a warning message “Not enough product quantities to sell”.
* Write appropriate constructor, methods to manage the sales and purchase processes.
* There should be a provision to view the available quantities.
* This should be a menu driven program, which allows to
  + Purchase Product
  + Sale Product
  + View Available Product Quantities
  + Exit
* Whenever sales take place, the system should process the quantities and then should show the available quantities.

Program - 4

* Write a program which is having a class InventoryManagement.
* System should manage the quantity of purchase and sales processes.
* Users should be able to purchase the product with different prices in the same object of Inventory Management - Use a dictionary, maintain a numerical index and store product price and product quantity against it.
* Whenever the product is sold, it will start deducting the product qty whichever entry is done first in the dictionary, so qty will be deducted as FIFO (First In First Out).
* Whenever product qty is deducted change the price of that index.
* Formula -
* Once the qty from each purchase is used make sure, it is not used again, so update the dictionary accordingly.
* Program should be able to handle the valuation.
* Formula for valuation - sum of price \* qty / sum of total qty.
* Write appropriate constructor, methods to manage the sales and purchase processes.
* There should be a provision to view the available quantities.
* This should be a menu driven program, which allows to
  + Purchase Product
  + Sale Product
  + View Available Product Quantities
  + Exit
* Whenever sales take place, the system should process the quantities and then should show the available quantities.

Program - 5

Write a program to read xml file.

Read parent tag and child tag inside that parent tag.

Read the attribute of that child tag and print the value of that attribute.

Use xml.dom.minidom library Example - from xml.dom.minidom import parse Copy the below xml text and save it as .xml file Use parse(path of xml file) to read xml file from path <odoo>

<template id="\_assets\_backend\_helpers" inherit\_id="web.\_assets\_backend\_helpers"> <xpath expr="//link" position="before">

<link rel="stylesheet" type="text/scss" href="/account/static/src/scss/variables.scss"/> </xpath>

</template>

<template id="assets\_backend" name="account assets" inherit\_id="web.assets\_backend"> <xpath expr="." position="inside">

<link rel="stylesheet"

href="/account/static/src/css/account\_bank\_and\_cash.css"/> <link rel="stylesheet" href="/account/static/src/css/account.css"/> <link rel="stylesheet" type="text/scss" href="/account/static/src/scss/account\_reconciliation.scss"/> <link rel="stylesheet" type="text/scss" href="/account/static/src/scss/account\_journal\_dashboard.scss"/> <link rel="stylesheet" type="text/scss" href="/account/static/src/scss/account\_dashboard.scss"/> <link rel="stylesheet"

href="/account/static/src/scss/section\_and\_note\_backend.scss"/>

<script type="text/javascript"

src="/account/static/src/js/reconciliation/reconciliation\_action.js"/>

<script type="text/javascript"

src="/account/static/src/js/reconciliation/reconciliation\_model.js"/>

<script type="text/javascript"

src="/account/static/src/js/reconciliation/reconciliation\_renderer.js"/>

<script type="text/javascript"

src="/account/static/src/js/reconciliation/tour\_reconciliation.js"/>

<script type="text/javascript"

src="/account/static/src/js/account\_payment\_field.js"/>

<script type="text/javascript" src="/account/static/src/js/bank\_statement.js"/>

<script type="text/javascript"

src="/account/static/src/js/section\_and\_note\_fields\_backend.js"/>

<script type="text/javascript" src="/account/static/src/js/tour.js"/> </xpath>

</template>

<template id="assets\_frontend" name="account assets" inherit\_id="web.assets\_frontend"> <xpath expr="." position="inside">

<script type="text/javascript" src="/account/static/src/js/account\_portal\_sidebar.js"/> </xpath>

</template>

<template id="qunit\_suite" name="account\_reconciliation tests"

inherit\_id="web.qunit\_suite">

<xpath expr="//script[contains(@src, '/web/static/tests/views/kanban\_tests.js')]" position="after">

<script type="text/javascript"

src="/account/static/tests/reconciliation\_tests.js"/>

<script type="text/javascript"

src="/account/static/tests/account\_payment\_field\_tests.js"/>

</xpath>

</template>

</odoo>

Program - 6

Create a program that read a .csv file and print its data in dictionary Data is about Sales Order

The dictionary should have SO number as key and its values as dictionaries

**{SO001 :**

**{‘customer’ : {‘name’:<customer name>,‘address 1’ : <value of address1>,’address 2’: <value of address 2>, ‘city’ :<city>, ‘country’: <country>},**

**’Products’:[**

**{‘sku’:<sku>,’price’:<price>,’qty’:<qty>},**

**{‘sku’:<sku>,’price’:<price>}**

**]**

**}**

**}**

There should be a provision in the program where code of country is converted to country’s full name and only full name of country should be stored in a dictionary. Use import csv library for managing csv task

| Order No | Customer | SKU | Qty | Price | Address1 | Address2 | Zipcode | City | Country |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
| SO001 | Mohit Patel | STARC | 1 | 25 | Address | Address2 | 67806 | Amritsar | USA |
|  |  | D1735 |  |  | 0002 | 0001 |  |  |  |
|  |  | 01 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO001 | Mohit Patel | STARC | 2 | 30 | Address | Address2 | 67806 | Amritsar | USA |
|  |  | D1735 |  |  | 0002 | 0001 |  |  |  |
|  |  | 02 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | MALTA | 7 | 45 | Address | Address2 | 67806 | Sydney | AU |
|  |  | F10451 |  |  | 0011 | 0030 |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO001 | Mohit Patel | STARC | 5 | 87 | Address | Address2 | 28777 | Amritsar | USA |
|  |  | D1735 |  |  | 0002 | 0001 |  |  |  |
|  |  | 03 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO004 | Ankit Batra | STARC | 6 | 88 | Address | Address2 | 22589 | Hamburg | ES |
|  |  | D1735 |  |  | 0005 | 0012 |  |  |  |
|  |  | 01 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | 7PLUS | 5 | 89 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | F17320 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | 7PLUS | 6 | 90 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | F17320 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | 7PLUS | 4 | 91 | Address | Address2 | 1720 | New York | UK |
|  |  | F17320 |  |  | 0008 | 0016 |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | 7PLUS | 3 | 93 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | F17320 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |



| SO003 | Ketan Desai | MALTA | 1 | 94 | Address | Address2 | 67806 | Sydney | AU |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | F10451 |  |  | 0011 | 0030 |  |  |  |
|  |  | 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO004 | Ankit Batra | MALTA | 4 | 95 | Address | Address2 | 22589 | Hamburg | ES |
|  |  | F10451 |  |  | 0005 | 0012 |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | MALTA | 2 | 97 | Address | Address2 | 67806 | Sydney | AU |
|  |  | F10451 |  |  | 0011 | 0030 |  |  |  |
|  |  | 3 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO005 | Prashant | MALTA | 3 | 98 | Address | Address2 | 22589 | Hamburg | IT |
|  | Singh | F10451 |  |  | 0015 | 0015 |  | Sülldorf |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | ASTN1 | 6 | 99 | Address | Address2 | 1720 | New York | UK |
|  |  | 73502 |  |  | 0008 | 0016 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | ASTN1 | 4 | 100 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | 73502 |  |  | 0006 | 0007 |  | hel |  |
|  |  |  |  |  |  |  |  |  |  |
| SO004 | Ankit Batra | ASTN1 | 5 | 102 | Address | Address2 | 22589 | Hamburg | ES |
|  |  | 73502 |  |  | 0005 | 0012 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | MSCE1 | 7 | 104 | Address | Address2 | 1720 | New York | UK |
|  |  | 83104 |  |  | 0008 | 0016 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO004 | Ankit Batra | MSCE1 | 7 | 105 | Address | Address2 | 22589 | Hamburg | ES |
|  |  | 83104 |  |  | 0005 | 0012 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | MSCE1 | 5 | 106 | Address | Address2 | 1720 | New York | UK |
|  |  | 83105 |  |  | 0008 | 0016 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | MSCE1 | 6 | 108 | Address | Address2 | 1720 | New York | UK |
|  |  | 83109 |  |  | 0008 | 0016 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO010 | Ajit Kapoor | STARC | 2 | 112 | Address | Address2 | 28777 | Peru | IT |
|  |  | F18280 |  |  | 0029 | 0035 |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | STARC | 4 | 113 | Address |  | 67806 | Sydney | AU |
|  |  | F18280 |  |  | 0011 |  |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | 7PLUS | 5 | 114 | Address |  | 67806 | Sydney | AU |
|  |  | F17320 |  |  | 0011 |  |  |  |  |
|  |  | 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO010 | Ajit Kapoor | 7PLUS | 4 | 118 | Address | Address2 | 28778 | Peru | IT |
|  |  | F17320 |  |  | 0029 | 0035 |  |  |  |
|  |  | 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | STARC | 2 | 134 | Address | Address2 | 1720 | New York | UK |
|  |  |  |  |  |  |  |  |  |  |



|  |  | E17290 |  |  | 0008 | 0016 |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | STARC | 4 | 136 | Address | Address2 | 67806 | Sydney | AU |
|  |  | E17290 |  |  | 0011 | 0030 |  |  |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO006 | Mihir Tanna | STARC | 7 | 137 | Address | Address2 | 1720 | New York | UK |
|  |  | E17290 |  |  | 0008 | 0016 |  |  |  |
|  |  | 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO003 | Ketan Desai | STARC | 1 | 138 | Address | Address2 | 67806 | Sydney | AU |
|  |  | E17290 |  |  | 0011 | 0030 |  |  |  |
|  |  | 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | 7PLUS | 5 | 141 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | F17450 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | MALTA | 6 | 144 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | E15030 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO002 | Rajesh | MALTA | 2 | 146 | Address | Address2 | 67806 | Dörrmosc | DE |
|  |  | E15030 |  |  | 0006 | 0007 |  | hel |  |
|  |  | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| SO001 | Mohit Patel | STARC | 4 | 45 | Address | Address2 | 1720 | Amritsar | USA |
|  |  | D1735 |  |  | 0002 | 0001 |  |  |  |
|  |  | 04 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

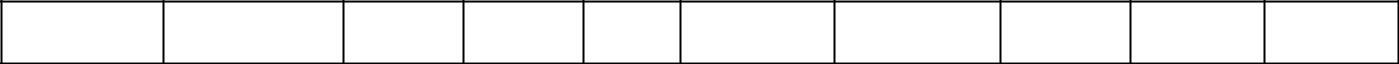
Program - 7

Create a program that write a .csv file

From Program 6 read the .csv file and create a dictionary and after that write the values from that dictionary to .csv file.

Read the dictionary and write in the “.csv” file

Following will be the headers



Order No

Customer

SKU

Qty

Price Address1 Address2

Zipcode City

Country

Program - 8

Create a program that write a .csv file

From Program 6 read the .csv file and create a dictionary and after that write the values from that dictionary to .xls file.

Read the dictionary and write in the “.xls” file

Following will be the headers

| Order No | Customer | SKU | Qty | Price | Address1 | Address2 | Zip Code | City | Country |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Using following import

from openpyxl import Workbook

Program - 9

Write a program which will handle the Sales Transaction This will be a menu driven program.

It should be able to store the product details such as product name, product unit price, product cost, product\_type(stockable, consumable, service), stock - separate dictionary

* Generate the unique sku automatically as follows PRD1, PRD3, PRD4...​*PRDn*
* Use sku as the key
* There will be a "create product" method.

There should be a separate method which will do the preparation of a dictionary of the product and return the prepared values which will be used to add a new product to the dictionary.

The create method of product should return the newly created index(which is sku in our case)

There should be provision to increase stock of the product There should be Sales process where sales order is generated

SO order no should be automatically generated as SO1, SO2,....SO10

SO should contain multiple products from product which are already created

It should show the list of products and one can select from the list, similar to menu driven, so that one can enter more than product to the sales order. If the same product is selected add the quantity, instead of creating new dict. One should enter the sku which will inform the program which product is to be added to the order. If the given SKU is not found in the product list, a warning is to be raised.

This inner dictionary of product should contain array of dictionary of product containing - product sku, product qty, product unit price, subtotal

Sales Order dictionary will contain all the orders with SO1 as key and value will be -customer detail(customer code)

-all the products in that sales order as dictionary

-total of the order

-state of the order - possible values will be ['draft','confirm',’done’,cancel'] -default state will be 'draft'

-Order date

There should be a provision to confirm order

There should be a provision to cancel the order

Only confirmed order can be marked as ‘done’.

Confirmed order can be cancelled.

Once the order is cancelled it cannot be reverted to confirm state directly, it should be first set to draft and then to confirm state

Done order cannot be changed to cancelled or confirm or draft

There should be provision to enter the Customer details (unique customer code, customer

name, customer email, customer phone) -

{cust\_1:{‘name’:<value>,’email’:<value>,’phone’:<value>}}

{cust\_1:{‘address1’:<value>,’address2’:<value>,‘city’:<value>,’state’:<value>,’country’:<value

>}}

There should be provision to enter the Address belonging to specific customer (customer code, Address1, Address2, City, Zipcode, State, Country)

The customer code in customer address must be from the customer details, otherwise warning is to be raised.

Order should contain the details of the customer - Customer code only, rest of the details should be fetched using the customer code and used in printing only Print the specific sales order as follows

Order No : <SO number>

Order Date : <order date>

Order Status : <Order status>

Customer : <cust\_code>, <customer name>

<phone>

<email>

<address 1>

<address 2>

<city>, <state>

<zipcode>

<country>

Product Name Product Price Product Quantity Subtotal

=========================================================

| Xyz | 88 | 7 | 2342.33 |
| --- | --- | --- | --- |
| Abc | 22 | 3 | 546.56 |

Order Total - 2334908.00

Edit:

Try to create a method named “prepare\_order\_lines” which only returns a prepared order line dictionary, one at a time. One dictionary at a time will be returned from it. So if someone is entering 3-4 products in an order, it will be called that many times and one dictionary per product will be returned. You can add more methods to support “prepare\_order\_lines” if you want to. “prepared\_order\_lines” will be called from the “prepare\_sales\_order” method. But if you have some other logic then you can do it. Currently product code, product unit price,

product qty, subtotal were added in each order line, add “state” to it with default value as “draft”.

Currently there is one menu to change the state of the order, instead convert it to following menus and separate methods for each process.

* Confirm Order
* Cancel Order
* Set To Draft
* Set to Done

When order is confirmed, change the state of each order line to “Confirm” and then change the state of order to “Confirm”.

Program - 11

[[2,4,5],[4,6,8],[3,6,9]]

Considering the above list, we need to have a sum of all the values of all the inner lists.

* Find the sum of only even numbers
* Find the sum of all the numbers from all the inner lists Don’t use a loop.

Program - 12

emp\_dict = {

101:{ 'name': 'Anupriya Roy',

'depart\_id':1,

'attendances':[{'date':1, 'hours':[3.5,4.5]},{'date':2, 'hours':[3.2,4.5]},{'date':3, 'hours':[3.2,4.6]},

{'date':4, 'hours':[3.0,4.5]},{'date':5, 'hours':[2.5,4.5]},{'date':6,

'hours':[1.5,4.5]},

{'date':7, 'hours':[2,3]},{'date':8, 'hours':[0,4.5]},{'date':9, 'hours':[2,3.5]}, {'date':10, 'hours':[4,3.5]}],

'leaves':[{'date':7, 'no\_of\_hours':1.5},{'date':7, 'no\_of\_hours':1.5},{'date':8, 'no\_of\_hours':3}]

},

102:

{ 'name': 'Kadambari Sharma',

'depart\_id':1,

'attendances':[{'date':1, 'hours': [0,4.5]},{'date':2, 'hours':[3.2,0]},{'date':3, 'hours':[3.2,4.6]},

{'date':4, 'hours':[1,4.5]},{'date':5, 'hours':[2.5,2]},{'date':6, 'hours':[1.5,1]}, {'date':7, 'hours':[2,4]},{'date':8, 'hours':[1,4.5]},{'date':9, 'hours':[2,2]}, {'date':10, 'hours':[2,3.5]}],

'leaves':[{'date':1, 'no\_of\_hours':3.5},{'date':2, 'no\_of\_hours':2},{'date':2, 'no\_of\_hours':2}]

},

103:

{ 'name': 'Abhishek Verma', 'depart\_id':1,

'attendances':[{'date':3, 'hours':[3.2,4.6]},{'date':4, 'hours':[1,4.5]},{'date':5,

'hours':[2.5,2]},

{'date':6, 'hours':[1.5,1]},{'date':7, 'hours':[2,4]},{'date':8, 'hours':[1,4.5]}, {'date':9, 'hours':[2,2]},{'date':10, 'hours':[2,3.5]}

],

'leaves':[{'date':1, 'no\_of\_hours':3},{'date':2, 'no\_of\_hours':2},{'date':2, 'no\_of\_hours':3}]

}

}

Find the following outputs - ​**Don’t use a loop. The only loop that will be used is the main dictionary loop of emp\_dict**

1.

Employee's total attendance and leave hours :

[

{'employee\_id': 101, ‘employee\_name’:’Anupriya Roy’,'total\_attendance\_hours': 66.5,

'total\_leave\_days': 6.0},

{'employee\_id': 102,,’employee\_name’:'Kadambari Sharma',

'total\_attendance\_hours': 49.0, 'total\_leave`\_days': 7.5},

{'employee\_id': 103,’employee\_name’:'Abhishek Verma', 'total\_attendance\_hours':

41.3, 'total\_leave\_days': 8}

]

2. Consider 8hrs of work duty, get the following output

Employee's day wise record in which they have not 7 hours attendance :

[

{101: {'date': [6, 7, 8, 9], 'total\_hrs': [6.0, 5, 4.5, 5.5], 'remaining\_hrs': [2.0, 3, 3.5, 2.5]}},

{102: {'date': [1, 2, 4, 5, 6, 7, 8, 9, 10], 'total\_hrs': [4.5, 3.2, 5.5, 4.5, 2.5, 6, 5.5, 4,

5.5], 'remaining\_hrs': [3.5, 4.8, 2.5, 3.5, 5.5, 2, 2.5, 4, 2.5]}},

{103: {'date': [4, 5, 6, 7, 8, 9, 10], 'total\_hrs': [5.5, 4.5, 2.5, 6, 5.5, 4, 5.5],

'remaining\_hrs': [2.5, 3.5, 5.5, 2, 2.5, 4, 2.5]}}

]

Program - 13

This program is an extension to Program-9.

Read about “Delivery Order” in reference to “Sales Order”. You can search on Google for this.

When the order is confirmed, create the delivery order(which will be a dictionary similar to sales order, product or customer, take this dictionary in sale order class). The key will be automatically generated as follows : DO1, DO2, DO3...DOn Each delivery order should be like as follows:

{DO1 : {‘sales\_order’ : SO1,

‘stock\_moves’:[{‘product\_code’:<value>, ‘product\_qty’:<value>,’state’:<value>}]

‘state’:<value>}}

Values for stock\_moves will be copied from order lines of related sales order

Values for “state” will be - draft(default value), done, cancelled.

Remove the menu of “Set to Done” order.

Add menus

* Validate Delivery Order
* Cancel Delivery Order

The changes from Program-9 in this program will be that quantity of product will be deducted from the master dictionary of products when the Delivery order is validated, not before that. That means not during order in draft or confirmed state.

Whenever Delivery order is validated, set the stock moves state to “done”, then mark the delivery order as done and then set the state of the related sales order as “Done”. Whenever Delivery order is cancelled, set the stock moves state to “cancel” and then set the state of delivery order as “Cancel”. Also, revert the quantities to the product dictionary. Don’t change the state of sales order automatically.

* If the current order state is confirmed and wants to cancel the order - Is there a delivery order created for this order, if yes is it in a cancel state then cancel the order. If it is not in a cancel state, then the system should not cancel the order and show the warning to cancel the delivery order first. If the delivery order is cancelled, then the system should allow you to cancel the order. If the delivery order is not created, then it should allow you to cancel the order. If the delivery order is in “done” state, then the order cannot be cancelled.
* If the current order state is confirmed and wants to set it to draft - Cannot directly set back to draft, first cancel the order and then bring it back to the draft state.
* If the current order state is cancelled and then changed to draft and then want to confirm it - Check if there is already any delivery order created for that sales order and is cancelled or not. As per the flow the delivery order must be cancelled, so now when the state is changed to confirm, it should create a new Delivery Order. In normal cases if it is getting confirmed for the first time, it will create the first Delivery Order, in case of cancel -> draft -> confirm it will keep on creating new delivery orders. It will check if the

