Fall 2015.

Point Of Sale Application

OOP244 Assignment V5.0 Final Project

-
-

CLASSES TO BE DEVELOPED

Date

PosIO

Item

Perishable

NonPerishable

PosSys

PROJECT DEVELOPMENT PROCESS

-
-
-
-
-

FILE STRUCTURE FOR THE PROJECT

```
TAX (0.13)              The tax rate for the goods
MAX_SKU_LEN (7)         The maximum size of an SKU code

MIN_YEAR (2000)         The min year used to validate year input
MAX_YEAR (2030)         The max year used to validate year input

MAX_NO_ITEMS (2000)      The maximum number of records in the data file.
```

```
#ifndef SICT_HeaderFileName_H__
#define SICT_HeaderFileName_H__


#endif
```

```
#ifndef SICT_POSSYS_H__
#define SICT_POSSYS_H__
```

MILESTONE 1: THE DATE CLASS

_dateOnly bool _dateOnly;

```
int _year;

int _mon;
int _day;

int _hour;
int _min;
int _readErrorCode;
```

```
          NO_ERROR    0  -- No error - the date is valid
          CIN_FAILED  1  -- istream failed on accepting information
          YEAR_ERROR  2  -- Year value is invalid
          MON_ERROR   3  -- Month value is invalid
          DAY_ERROR   4  -- Day value is invalid
          HOUR_ERROR  5  -- Hour value is invalid
          MIN_ERROR   6  -- Minute value is invalid
```

```
bool _dateOnly;
```

```
int value()const; (this function is already implemented and provided)
```

```
void errCode(int errorCode);

void set(int year, int mon, int day, int hour, int min);
```

NO_ERROR.

void set();

```
bool operator==(const Date& D)const;
bool operator!=(const Date& D)const;
bool operator<(const Date& D)const;
bool operator>(const Date& D)const;
bool operator<=(const Date& D)const;
bool operator>=(const Date& D)const;
```

operator<

this->value() D.value()

`int mdays()const;` (this function is already implemented and provided)

`void set();` (this function is already implemented and provided)

```
int errCode()const;
bool bad()const;
bool dateOnly()const;
void dateOnly(bool value);
```

```
std::istream& read(std::istream& is = std::cin);
```

(istr)

_readErrorCode CIN_FAILED

(istr)

_readErrorCode

(istr)

```
std::ostream& write(std::ostream& ostr = std::cout)const;
```

(ostr)

(istr)

MILESTONE 2: THE POSIO INTERFACE V1.0

fstream **iostream**

Pure virtual member functions (methods):

Linear:
Form:

Submission:

MILESTONE 3: THE ITEM CLASS

_sku:

_name:

_price

_taxed:

_quantity:

-
-
-
-
-

Copy Constructor;
See below:

```
-
- _
-   d
-   d
-    d
```

```
-
```

```
-          B  B B    B         B
```

R d : receives an integer and returns an integer.

R d : receives an integer and returns an integer.

R d :

MILESTONE 4: THE NONPERISHABLE AND PERISHABLE CLASSES

Part one: NonPerishable

ErrorMessage

ErrorMessage();

void clear();

bool isClear()const;

void message(const char* value);

const char* message()const;

_err

“N”

file

```
sku, name, price, taxed, quantity
```

```
N,1234,Candle,1.23,1,38<Newline>
```

N,”

Hint: create temporary variables of type double, int and string and read the fields one by one, skipping the commas. After each read, set the member variables using setter methods.

_err

_err

```
1234   |Candle              |   1.23| t |  38|    52.82|
```

SKU:
Name:
Price:
Taxed:
Quantity:
Cost:
One Bar **NO NEW LINE**

```
Name:
Candle
Sku: 1234
Price: 1.23
Price after tax: 1.39
```

```
Quantity: 38
Total Cost: 52.82 <Newline>

OR if not taxed

Sku: 1234
Name: Candle
Price: 1.23
Price after tax: N/A
Quantity: 38
Total Cost: 46.74 <Newline>
```

```
Non-Perishable Item Entry:
Sku: 1234<ENTER>
Name:
Candle<ENTER>
Price: 1.23<ENTER>
Taxed: y<Enter>
Quantity: 38<ENTER>
```

istr **fail**

_err

Invalid Price Entry
Invalid Taxed Entry, (y)es or (n)o
Invalid Quantity Entry

```
.setstate(ios::failbit);
```

Part Two: Perishable Class

Please note that the Perishable and NonPerishable classes are identical in logic. The only difference is that the Perishable class has one extra member variables that have to be received and printed (in addition to the variables in an Item).

- **_err**
-

P"

file

```
sku, name, price, taxed, quantity, expiry date
```

```
P,1234,4L Milk,3.99,0,2,2015/12/10<Newline>
```

load **P,**

_err
_err

Linear is true:

```
1234   |4L Milk             |   3.99|  p|   2|      7.98|
```

Sku:
Name:
Price:
Taxed:
Quantity:
Cost:
NO NEW LINE

Linear is false:

```
Name:
4L Milk
Sku: 1234
Price: 3.99
Price after tax: 4.51
Quantity: 2
Total Cost: 9.02
Expiry date: 2015/12/10 <Newline>
```

```
Name:
4L Milk
```

```
Sku: 1234
Price: 3.99
Price after tax: N/A
Quantity: 2
Total Cost: 7.98
Expiry date: 2015/12/10 <Newline>
```

```
Perishable Item Entry:
Sku: 1234<ENTER>
Name:
4L Milk<ENTER>
Price: 3.99<ENTER>
Taxed: n<ENTER>
Quantity: 2<ENTER>
Expiry date (YYYY/MM/DD) : 2015/12/10<ENTER>
```

istr **fail**

_err

Invalid Price Entry
Invalid Taxed Entry, (y)es or (n)o
Invalid Quantity Entry

```
.setstate(ios::failbit);
```

_err

CIN_FAILED: **Invalid Date Entry**

YEAR_ERROR: **Invalid Year in Date Entry**

MON_ERROR: **Invalid Month in Date Entry**

DAY_ERROR: **Invalid Day in Date Entry**

```
.setstate(ios::failbit);
```

MILESTONE 5: THE POSAPP CLASS

PosApp Class

Constructors and assignment operator overload

S d d d d

Copying and assignment

Private member variables (attributes)

d d

d d

P QR P

R

Private member functions (methods)

```
The OOPs Store
1- List items
2- Add Perishable item
3- Add Non-Perishable item
4- Update item quantity
5- Show Item
6- POS
0- exit program
> _
```

Data management member functions (methods)

d

```
    set readIndex to zero

open the file for reading (use ios::in)
if the file is in fail state, it means there is no file on the disk, then
  clear the failure
  close the file
  open the file for writing (ios::out) to create the file
  close thefile
otherwise (if the file is not in fail state)
  until reading fails loop:
     delete the item pointer at readindex. (not to have memory leak)
     read one character into the Id character
     if Id character is P
       Dynamically create a Perishable item and hold it in item pointer at readIndex
     if Id character is N
       Dynamically create a NFI item and hold it in item pointer at readIndex
     if either P or N is read
        skip the comma in the file
        load the data into the newly created item from the file
                                            (using its load method)
        add one to readindex
  continue the loop
set number of items to readIndex
close the datafile
```

d

d d

d T

```
>Start
Please enter the SKU: 9999
Not found!

>END
```

```
>Start
Please enter the SKU: 1234
Name:
Milk
Sku: 1234
Price: 3.99
Price after tax: N/A
Quantity: 2
Total Cost: 7.98
Expiry date: 2015/10/04

Please enter the number of purchased items: 5
Updated!

>END
```

    d   S   d

```
>Start
Perishable Item Entry:
Sku: abc
name:
abc
price: abc
Invalid Price Entry

>END
```

```
>Start
NonPerishable Item Entry:
Sku: 3456
Name:
Paper Cups
Price: 5.99
Taxed: y
Quantity: 40
Item added.
```

```
>END
```

```
>Start
 Row | SKU     | Item Name           | Price |TX |Qty |   Total |
-----|--------|--------------------|-------|---|----|---------|
   1 | 1234   |Milk                |   3.99|  p|   7|    27.93|
   2 | 2345   |Soap                |  23.45| t |   2|    53.00|
   3 | 3456   |Paper Cups          |   5.99| t |  40|   270.75|
-----^--------^--------------------^-------^---^----^---------^
Total Asset: $351.67

>END
```

Point Of Sale member functions (methods)

E

```
>Start
v--------------------------------------------------------v
| 2015/12/01, 11:30                                      |
| SKU     | Item Name          | Price |TX |Qty |   Total |
|--------|--------------------|-------|---|----|---------|
| 1234   |Milk                |   3.99|  p|   1|     3.99|
| 2345   |Soap                |  23.45| t |   1|    26.50|
| 3456   |Paper Cups          |   5.99| t |   1|     6.77|
| 2345   |Soap                |  23.45| t |   1|    26.50|
^--------^--------------------^-------^---^----^---------^
Total $63.76
```

>END

d E

qty

qty

```
void POS();
```

```
Pseudo:
while not done
   Ask for sku
   If sku is an empty string
      show bill and the function is done
   Search for sku in the items
   if found
      print the name only
      add it to the bill
   if not found
      print “Not found!”
End while
```

```
>START
Sku: 1234
v------------------->
| Milk
^------------------->
Sku: 1234
v------------------->
| Milk
^------------------->
Sku: 2345
Not found!
```

```
Sku: 3456
v------------------->
| Paper Cups
^------------------->
Sku: 4567
v------------------->
| Butter
^------------------->
Sku:
v--------------------------------------------------------v
| 2015/12/01, 12:03                                       |
| SKU     | Item Name           | Price |TX |Qty |   Total |
|--------|--------------------|-------|---|----|---------|
| 1234   |Milk                |   3.99|  p|   1|     3.99|
| 1234   |Milk                |   3.99|  p|   1|     3.99|
| 3456   |Paper Cups          |   5.99| t |   1|     6.77|
| 4567   |Butter              |   4.56| tp|   1|     5.15|
^--------^--------------------^-------^---^----^---------^
Total $19.90

>END
```

Public member function (method)

```
void run();
```

In a continuous loop run will display the menu and wait for user’s selection.

The following will happen upon user’s selection:
User selects 1:
Call listItems.

User selects 2:
Call addItem.

User selects 3:
Call addItem.

User selects 4:
Call updateQty.

User selects 5:
Prompt the user for receiving sku.
get the sku and searchItems() for it.
if found display it in non-linear format
if no found display “Not found!”.

User selects 6:
Call POS.

User selects 0:
Exit program printing “Goodbye!”.

```
>START
The OOPs Store
1- List items
2- Add Perishable item
3- Add Non-Perishable  item
4- Update item quantity
5- Show Item
6- POS
0- exit program
> 5

Please enter the SKU: 1234
v------------------------------------v
Name:
Milk
Sku: 1234
Price: 3.99
Price after tax: N/A
Quantity: 4
Total Cost: 15.96
Expiry date: 2015/10/04
^------------------------------------^

The OOPs Store
1- List items
2- Add Perishable item
3- Add Non-Perishable  item
4- Update item quantity
5- Show Item
6- POS
0- exit program
> 5

Please enter the SKU: 2345
Not found!

The OOPs Store
1- List items
2- Add Perishable item
3- Add Non-Perishable  item
4- Update item quantity
5- Show Item
6- POS
0- exit program
> 20
```

```
===Invalid Selection, try again===

The OOPs Store
1- List items
2- Add Perishable item
3- Add Non-Perishable  item
4- Update item quantity
5- Show Item
6- POS
0- exit program
> 0

Goodbye!!
>END
```

```
~fardad.soleimanoo/fp <ENTER>
```

Submission