CS202: Assignment 7(Templates)

Spring 2021

1 Problem Introduction

The primary objective of this assignment is to make you familiar with template function and template class in C++. On the way, you have to use the parsing techniques to read the csv file, parse the data to tokenize it and use the tokenized data to create objects. Finally, you need to create a template class to sort the records and write a template function to find the mode of the transaction dates and product prices. Now, let's talk about the problem in more detail.

You are provided with a csv file "SalesJan2009.csv" which contains several purchase transaction made during 2009. Each transaction contains information like Transaction Id, Date of Transaction, Product Name, Name, City and Country of the person making the transaction etc. Your first task is to read the file and parse it. After you parse each field of the file, use that information to create objects of class Transaction. This class is also provided to you. You just need to create objects by passing all those transaction information as parameters of the constructor of class Transaction.

The file "SalesJan2009.csv" contains several transaction, hence you have to create several objects of class Transaction(one for each record in the file). Next, the transaction objects should be populated in a vector. After that the transactions should be sorted by date and product price using the **sort** function from the **sorter** class.

2 Classes and Files

2.1 Transaction class

This is a class provided to you as transaction.h header file. Following are the members of the class

Transaction
-transactionId : int -transactionDate : Date -product : string -price : int -paymentType : string -name : string -city : string -state: string -country : string
+Transaction(Date, string,int,string,string,string,string,string) +getId():int +getPrice():int +getName():string +getDate():Date +getProduct():string

The constructor and functions are self explanatory and they have been already implemented. You just have to use this class to create objects of Transaction type and populate the vector.

2.2 Date class

Date
-year:int -month:int -day:int -hour:int -minute:int
+Date(int,int,int,int,int) +operator>(Date&):bool +operator<<(ostream obj,Date d):ostream& friend +operator==(Date&):bool

Date class contains five private members to hold year, month, day, hour and second. Operator > has been overloaded to check for the later date. It also has a friend function where we have overloaded the insertion operator << to print out the date. Operator == has been overloaded to check if two dates have an equal day. All these function are already implemented.

2.3 Sorter class

sorter
-list:vector<T>* -ids:vector<int>
+sort():void +getListVector():vector<T>* +getIdsVector():vector<int>

This is the template class you need to implement by yourself. It takes either a vector of Transaction Date as a pointer and a vector of Transaction Id. You need to implement a sort function inside this class which sorts the **vector<Date>** and **vector<int>**(for product price) whichever comes from the main function. This is where the template class comes in. While sorting those vector, also sort the ids vector at the same time. After the sort function completes execution, both Date vector/Product Price vector and id vector should be sorted. Also, implement the getListVector() and getIDsVector() to return the list vector and ids vector.

2.4 Main file

Main file is where you need to implement some functions. Following are the functions inside the main file.

2.4.1 readFile

The prototype of this function is given below:

```
vector<Transaction> readFile(string fileName);
```

This function takes fileName as a parameter from the main function, reads the content of file "SalesJan2009.csv" and returns the result as a vector of Transactions. However, this function calls another function "Parse" to parse each line of the file to a Transaction object. readFile function has been implemented for you.

2.4.2 Parse

The prototype of this function is given below:

```
Transaction parse(string record);
```

It takes each line read by the function readFile as a parameter and parse them to create object of Transaction type. Each record of the file looks like below:

Transaction Id	Transaction Date	Product	Price	Payment Type	Name	City	State	Country
1	1/2/09 6:17	Product 1	1299	MasterCard	Carolina	Basildon	England	United Kingdom

This is the function you should implement. Tokenize each fields and further parse the Transaction Date field to create an object of Date type. Finally, create a Transaction object and return back to readFile function.

2.5 findModeTransaction

The prototype of this function is:

```
int findModeTransaction(vector<T> records,vector<int>  ids);
```

You are required to implement this function too. Once you sort the vector of transaction Dates or vector of Transaction Id this function should take these as a parameter and find the mode of the transaction Dates or product price. Mode of the transaction Dates is the Date where the most transaction occurred. You just have to check the day of the transaction. Any day that gets the most transactions should be returned. == operator for the date object has been overloaded to check if two transaction happened on the same day. Similarly, mode of the product price means the price which has occurred the most. **Inside this function, also print the content of the vector**. **Implement the logic of finding the mode without using any set or hashmap. Sorted vector of dates or product prices makes it easier to find mode without using additional data structure.** This function returns the mode of the transaction date or the product prices.

2.5.1 Main function

This is the function where the program starts. Let's go through the sequence of execution inside this function. You start by calling the **readFile** function with the filename as a parameter. It returns a vector of transaction objects. Next, it creates three vector dates, ids and prices which stores all the dates, id and prices of the transactions. Then, it creates an object of Sorter class which you use to call the sort function to sort the vector of transaction dates and then the vector of product prices. After each sort call, you call the template function findModeTransaction with sorted vector as a parameter. This function returns the mode of the transaction date or the product prices. Finally, the mode date and the product name corresponding to the mode price is printed.

3 Files included in your assignment

- main.cpp
- transaction.h
- date.h
- sorter.h
- SalesJan2009.csv
- sample_output

4 Compiling and submitting

Compile the program:

```
g++ main.cpp -o objFile
```

transaction.h, date.h and sorter.h are included inside main.cpp. Include main.cpp,transaction.h,date.h,sorter.h, SalesJan2009.csv, in your submission.

Include all those code files in a zip file <youname_hw7>.zip and submit.

5 Problem introduction video

Here is the video link for problem introduction: `https://www.youtube.com/watch?v=SUu4pPZ2KGg`