Object-Oriented Design and Programming (CSC319)

State University of New York at Plattsburgh

Project #2

Submission: one per team. Upload a single zipped file, with both files.

Final Deadline: see Moodle.

Grading:

1. Class diagram(using OO concept in your design/implementation): 20%
2. Code Implementation: implementation of the class diagram in C++: 80%
3. Bonus: 10%, if your final submission is 7 days before the deadline.

If you do not participate to the code implementation, this component of your grade will be lowered.

Deliverables/deadlines:

Final deliverable: a single zipped file containing:

- one file containing your Class diagram(pdf),
- a subdirectory, with your sources files. Each class

Specification:

Your program must process a set of transaction regarding sale. For each transaction, sale representative (aka salerep) are awarded points. We calculate their points based on a % of the transaction. Each salerep belongs to only a territory. A territory can either be a normal territory, or a premium territory. For a normal territory, the salerep receives only the standard % of the transaction, but for a premium territory, it receives 50% more points when points are added, and the normal % when points are subtracted.

You must write a main class in project2.cpp, that will process the transactions and based on it, it must generates updates in a file and multiple processing summaries.

Your program will receive as input 4 filename, territory.txt and transaction.txt, and salerep.txt for input and output and territory_output.txt.

territory.txt: File format: <territoryid>,<NORMAL|PREMIUM>
transaction.txt: File format: <trxid>,<salerepid>,<transaction type>,<amount>

salerep.txt: File format: <salerepid>,<territoryid>,<amount>

There is one file for the salereps as input/output (mode= "in+binary").
The file contains pre-existing amount per salereps which are running totals.
The transactions must be applied to the starting amount per salerep.
Once processing is completed, you must update the current content of the file (random I/O, seek then write)

Your program calculates based on a set of transactions, the amount of sales per salerep and per territory.

Valid assumptions:

- All amount are long (no decimal).
- Every ids in the system should be valid as input (no need for validation).

A fourth arguments, is territory_output.txt file.

territory_output.txt: <territoryid, amount> using this format: 00001,0002000

Your program with write to the file territory_output.txt, in descending order of amount, all the territories.

Command line:
Project2 <territory input filename> <salerep input/output filename> <transaction input filename> <territory output filename> <territory output filename>

Once the program is done the processing, it must produce an output on the console (cout) listing in descending order of amount, all the salerep such as:
salerepid, amount ex: 0020,0007000. You must follow the mask in the example.

To process the parameter passed as argument to a program:

```
int main(int argc, char *argv[])
{
	cout << argc << endl; //argument counts
	cout << argv[0] << endl; //1st argument
	cout << argv[1] << endl; //2nd argument
	cout << argv[2] << endl; //3rd argument
}
```

TRANSACTION PROCESSING

Each transaction has different rule in term of calculating the % of the transaction's amount which must be attributed to the salerep vs the territory.

Transaction Type & Attribution Rule

	Transaction Type	Territory	Salerep %	Operation	Comment
		%	%		
SALE	1	100	110	Added	
VALUEADDED	2	100	110	Added	Value added goods and services have very high marging.
CREDIT	3	100	100	Substracted	
CANCEL	4	100	125	Substracted	It is the duty of a salerep to minimize invoice cancellation as they may have legal incidence
PROMO	5	100	0	Substracted	Promo are given at the corporate level, therefore the salerep quota are not affected by it
DISCOUNT	6	100	110	Substracted	To minimize the likelyhood of a salerep giving discount
INTER-TERRITORY	7	0	75	Added	Added. Inter-territory sales are not accounted toward the territory total sales as the saler wil have it as a SALE

Example:
if a transaction.type == TRANSACTION.SALE with an amount of 200, the territory will have 200 * 1.00 (100%) and the salrerep will have 200 * 1.10 (110%) added to their amount.

if a transaction.type == TRANSACTION.PROMO with an amount of 200, 200 will be subtracted from the territory (100%) but 0 to the salerep (0%)

if a transaction.type == TRANSACTION.INTER-TERRITORY with an amount of 200, 0 will be added from the territory (0% of 200) but 150 to the salerep (75% of 200)