

```

1  ///Marc Pfeiffer
2  ///Assignment 1 - Accounts/Receivable
3
4  #include <iostream>
5  #include <fstream>
6  #include <string>
7
8  using namespace std;
9
10 class Customer{
11
12 public:
13
14     Customer();
15     friend ifstream& operator >>(ifstream &masterFile, Customer& info);
16     int getNum(){return number;}
17     string getName(){return name;}
18     double getBal(){return previousBalance;}
19
20 private:
21
22     int number;
23     string name;
24     double previousBalance;
25
26 };
27
28 ///Constructor for Customer
29 Customer::Customer(){
30
31     number=0;
32     name = "";
33     previousBalance = 0.00;
34 }
35
36 ///Overloaded in-file Operator to read in each line from master file.
37 ifstream & operator >> (ifstream &masterFile, Customer &info){
38
39     masterFile>>info.number>>info.name>>info.previousBalance;
40     return masterFile;
41 }
42
43 class Record{
44
45 public:
46
47     Record();
48     friend ifstream& operator >>(ifstream &transactionFile, Record& accounts);
49     int getCustNumb(){return custNumb;}
50     int getTransNumber(){return transNumb;}
51     double getAmount(){return (price*quantity);}
52     char getTransType(){return transType;}
53     string getItem(){return object;}
54
55 private:
56
57     int custNumb;
58     int transNumb;
59     char transType;
60     string object;
61     double price;
62     int quantity;
63 };
64
65 ///Constructor for Record class
66 Record::Record(){

```

```

67     custNumb=transNumb=quantity = 0;
68     transType='a';
69     object="";
70     price=0.00;
71 }
72
73
74 ///Overloaded in-file Operator reads in data from transaction file
75 ifstream & operator >>(ifstream &transactionFile, Record &accounts){
76
77     transactionFile >> accounts.custNumb>>accounts.transNumb>> accounts.transType;
78
79     if (accounts.transType=='o'){
80         transactionFile>>accounts.object>>accounts.price>>accounts.quantity;
81     }
82     else{
83         transactionFile>>accounts.price;
84         accounts.quantity = 1;
85     }
86
87     return transactionFile;
88 }
89
90
91 int main()
92 {
93     Record accounts;
94     Customer info;
95     double balance=0.00;
96     int checker=0;
97
98     ifstream transactionFile;
99     ifstream masterFile;
100    ofstream outfufFile("output.txt");
101    outfufFile.precision(9);
102
103    transactionFile.open("transfile.txt");
104
105    transactionFile>>accounts;
106
107    /// loops till the end of transaction file
108    while (!transactionFile.eof()){
109
110        ///if line started with a new number that was not used before
111        if(accounts.getCustNumb() != checker){
112
113            checker = accounts.getCustNumb();
114
115            masterFile.open("masterfile.txt");
116            masterFile>>info;
117
118            /// searches for correct customer info
119            while(info.getNum() != accounts.getCustNumb()){
120
121                masterFile>>info;
122            }
123
124            balance = info.getBal();
125            masterFile.close();
126
127            outfufFile<<"CUSTOMER: "<<info.getName()<<" "
128            <<"\tCUSTOMER ID: "<<info.getNum()
129            <<"\n\n\t\tPREVIOUS BALANCE $"<<info.getBal()<<"\n\n";
130        }
131
132        ///continues printing out different transactions for same customer

```

```

133     ///until the customer id changes
134     while(accounts.getCustNumb() == checker){
135
136         outpufFile<<accounts.getTransNumber()<<"\t";
137
138         /// if transaction is an order
139         if(accounts.getTransType() == 'o'){
140
141             outpufFile<<accounts.getItem()<<"\t\t"<<accounts.getAmount()<<endl;
142             balance= balance + accounts.getAmount();
143         }
144
145         /// if transaction is a payment
146         else{
147
148             outpufFile<<"PAYEMENT\t$"<<accounts.getAmount()<<endl;
149             balance = balance - accounts.getAmount();
150         }
151         transactionFile>>accounts;
152     }
153
154     outpufFile<<"\n\t\t\tBalance Due: $"<<balance<<"\n\n";
155
156 }
157
158 return 0;
159 }

```