Jairun Diemert

Project3

Step 1

```
//File: Transaction.h
//Author: Jairun Diemert
//Class: COP3014
//Project: 3
//Description: This is the header file for the Transaction class.

#ifndef TRANSACTION_H
#define TRANSACTION_H

#include <iostream>
#include <string>

using namespace std;

class Transaction {
public:
    Transaction();
    Transaction(int h, int m, int s, string n, int c, int p);
    ~Transaction();

    int GetH();
    void SetH(int h);

    int GetM();
    void SetM(int m);

    int GetS();
    void SetS(int s);

    int GetT();
    void SetT();

    string GetN();
    void SetN(string n);

    int GetC();
    void SetC(int c);

    int GetP();
    void SetP(int p);

    int GetE();
    void SetE();

private:
    int hour;
    int min;
    int sec;
    int trans;
    string name;
    int cost;
    int price;
    int earnings;
};


#endif
```

Step 2

```cpp
//File: Transaction.cpp
//Author: Jairun Diemert
//Class: COP3014
//Project: 3
//Description: This is the member function file for Transaction.

#include <iostream>
#include <stdio.h>
#include <string>
#include <stdlib.h>
#include <time.h>

using namespace std;

#include "Transaction.h"

Transaction::Transaction() {
    srand(time(NULL));
    hour = -1;
    min = -1;
    sec = -1;
    trans = -1;
    name = "Empty";
    cost = 0;
    price = 0;
    earnings = 0;
};
Transaction::Transaction(int h, int m, int s, string n, int c, int p) {
    hour = h;
    min = m;
    sec = s;
    trans = rand() % 1000 + 1;
    name = n;
    cost = c;
    price = p;
    earnings = price - cost;
};
Transaction::~Transaction() {
};

int Transaction::GetH() {
    return hour;
};
void Transaction::SetH(int h) {
    hour = h;
};

int Transaction::GetM() {
    return min;
};
void Transaction::SetM(int m) {
    min = m;
};

int Transaction::GetS() {
    return sec;
};
void Transaction::SetS(int s) {
```

```
58  void Transaction::SetS(int s) {
59      sec = s;
60  };
61
62  int Transaction::GetT() {
63      return trans;
64  };
65  void Transaction::SetT() {
66      trans = rand() % 1000 + 1;
67  };
68
69  string Transaction::GetN() {
70      return name;
71  };
72  void Transaction::SetN(string n) {
73      name = n;
74  };
75
76  int Transaction::GetC() {
77      return cost;
78  };
79  void Transaction::SetC(int c) {
80      cost = c;
81  };
82
83  int Transaction::GetP() {
84      return price;
85  };
86  void Transaction::SetP(int p) {
87      price = p;
88  };
89
90  int Transaction::GetE() {
91      return earnings;
92  };
93  void Transaction::SetE() {
94      earnings = price - cost;
95  };
96
```

Step 3

```cpp
//File: Main.cpp
//Author: Jairun Diemert
//Class: COP3014
//Project: 3
//Description: This is the Main file.
#include <iostream>
using namespace std;

#include "Transaction.h"

int main() {


    while (!end) {
        system("CLS");

        char option = '0';

        cout << endl;
        cout << "Please choose an option below" << endl;
        cout << "-----------------------------" << endl;
        cout << "1. Enter a transaction." << endl;
        cout << "2. Print transaction summary." << endl;
        cout << "3. Void a transaction." << endl;
        cout << "4. End Program." << endl;
        cout << ": ";
        cin >> option;
        switch (option) {
        case '1':

        case '2':

        case '3':

        case '4':
            end = true;
            break;
        default:
            system("CLS");
            cout << endl<< "\tYour entry was invalid please try again." << endl << endl;
            system("pause");
            break;
        }
    }
    return 0;
}
```

Step 4

```cpp
Transaction *sale[10];
for (int i = 0; i < 10; i++) {
    sale[i] = new Transaction;
}
//Test data if needed
/*sale[0] = new Transaction(1, 1, 1, "one", 1, 2);
sale[1] = new Transaction(2, 2, 2, "two", 2, 3);
sale[2] = new Transaction(3, 3, 3, "three", 3, 5);
sale[3] = new Transaction(20, 4, 4, "four", 4, 23);
sale[5] = new Transaction(6, 6, 6, "six", 6, 63);
sale[6] = new Transaction(12, 7, 7, "seven", 7, 25);
sale[7] = new Transaction(8, 8, 8, "eight", 8, 78);
sale[8] = new Transaction(9, 9, 9, "nine", 9, 2);
sale[9] = new Transaction(12, 9, 9, "ten", 9, 0);*/
```

Step 5 & Step 6

```
case '1':
    int i;
    system("CLS");
    for (i = 0; i < 10; i++) {
        if (sale[i]->GetH() == -1) {
            cout << endl;
            cout << "Enter the Hour using 0-23 integer notation: ";
            cin >> h;
            sale[i]->SetH(h);
            cout << endl << endl;

            cout << "Enter the Min using 0-59 integer notation: ";
            cin >> m;
            sale[i]->SetM(m);
            cout << endl << endl;

            cout << "Enter the second using 0-59 integer notation: ";
            cin >> s;
            sale[i]->SetS(s);
            cout << endl << endl;

            sale[i]->SetT();

            cout << "Enter the name using lower case letters with no spaces: ";
            cin >> n;
            sale[i]->SetN(n);
            cout << endl << endl;

            cout << "Enter the cost using rounded integer notation: ";
            cin >> c;
            sale[i]->SetC(c);
            cout << endl << endl;

            cout << "Enter the price using rounded integer notation: ";
            cin >> p;
            sale[i]->SetP(p);
            cout << endl << endl;

            sale[i]->SetE();
            cout << "\tTransaction added." << endl << endl;
            break;
        }
    }
    if (i > 9) {
        cout << endl << "\tThere is no more space for new transactions." << endl << endl;
    }
    system("pause");
    break;
```

```
105             case '2':
106                 end2 = false;
107                 while (!end2) {
108                     system("CLS");
109 
110                     char option2 = '0';
111 
112                     cout << endl;
113                     cout << "Please choose an option below" << endl;
114                     cout << "-----------------------------" << endl;
115                     cout << "1. Sort summary by trans#." << endl;
116                     cout << "2. Sort summary by earnings." << endl;
117                     cout << "3. Sort summary alphabetically." << endl;
118                     cout << ": ";
119                     cin >> option2;
120                     switch (option2) {
121                     case '1':
122                         system("CLS");
123                         cout << endl;
124                         for (int i = 0; i < 10; i++)
125                         {
126                             for (int j = i + 1; j < 10; j++) {
127                                 if (sale[j]->GetT() < sale[i]->GetT()) {
128                                     temp = *sale[i];
129                                     *sale[i] = *sale[j];
130                                     *sale[j] = temp;
131                                 }
132                             }
133                         }
134                         cout << endl;
135                         end2 = true;
136                         break;
137                     case '2':
138                         system("CLS");
139                         cout << endl;
140                         for (int i = 0; i < 10; i++)
141                         {
142                             for (int j = i + 1; j < 10; j++) {
143                                 if (sale[j]->GetE() < sale[i]->GetE()) {
144                                     temp = *sale[i];
145                                     *sale[i] = *sale[j];
146                                     *sale[j] = temp;
147                                 }
148                             }
149                         }
150                         cout << endl;
151                         end2 = true;
152                         break;
153                     case '3':
154                         system("CLS");
155                         cout << endl;
156                         for (int i = 0; i < 10; i++)
157                         {
158                             for (int j = i + 1; j < 10; j++) {
159                                 if (sale[j]->GetN() < sale[i]->GetN()) {
160                                     temp = *sale[i];
161                                     *sale[i] = *sale[j];
162                                     *sale[j] = temp;
```

```
                end2 = true;
                break;
            default:
                cout << "\tYour entry was invalid please try again." << endl;
                break;
            }
        }

        system("CLS");
        saleCounter = 0;
        sumEarnings = 0;
        cout << endl;
        cout << "(TRANS#) HOUR:MIN:SEC NAME[ PRICE - COST = EARNINGS ]" << endl;
        cout << "-------------------------------------------------" << endl;
        for (int i = 0; i < 10; i++) {
            if (sale[i]->GetH() != -1) {
                cout << "\t(" << sale[i]->GetT() << ") ";
                cout << sale[i]->GetH() << ":";
                cout << sale[i]->GetM() << ":";
                cout << sale[i]->GetS() << " ";
                cout << sale[i]->GetN() << " [ $";
                cout << sale[i]->GetP() << " - $";
                cout << sale[i]->GetC() << " = $";
                cout << sale[i]->GetE() << " ]" << endl;
                sumEarnings = sumEarnings + sale[i]->GetE();
                saleCounter++;
            }
        }
        if (saleCounter != 0) {
            cout << endl << endl << "\tThe total profit for the day is: $" << sumEarnings << endl << endl;
            cout << "\tThe average profit for each item is: $" << sumEarnings / (saleCounter) << endl << endl;
        }
        if (saleCounter == 0) {
            cout << "\tThere are no transactions to show." << endl << endl;
        }
        system("pause");
        break;
```

```
    case '3':
        system("CLS");
        int k;
        cout << endl;
        cout << "Enter the item# you wish to void: ";
        cin >> transNum;
        for (k = 0; k < 10; k++) {
            if (sale[k]->GetT() == transNum) {
                h = -1;
                sale[k]->SetH(h);
                cout << endl << "\tTransaction " << transNum << " voided." << endl << endl;
                break;
            }
        }
        if (k == 10) {
            cout << endl << "\tThe trans# you entered is invalid." << endl << endl;
        }
        system("pause");
        break;
    case '4':
        end = true;
        break;
    default:
        system("CLS");
        cout << endl<< "\tYour entry was invalid please try again." << endl << endl;
        system("pause");
        break;
    }
  }
  return 0;
}
```

Step 7 TEST……..Done