Minwossi Zerbo

CIS 17A Project 1

Manager tracking software.

04/27/15

Project size: 232 lines

Variables: Up to 20

**Summary**

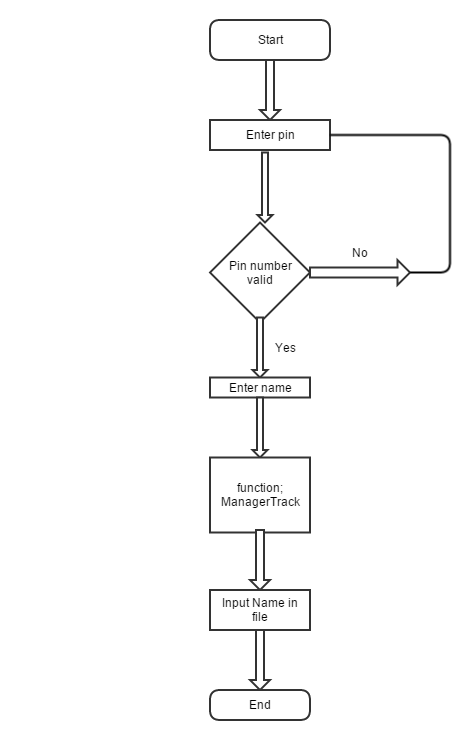
The program i came up with is a software that allows a store, a supermarket or any kind of market too keep track of their inventory and receive a delivery sheet every time the get items. And the program also allows the manager to calculate the payment of his employees and keep track of it. All these was made in a 232 lines program including structures, array of structures, pointers, dynamic arrays, and writing into a file.

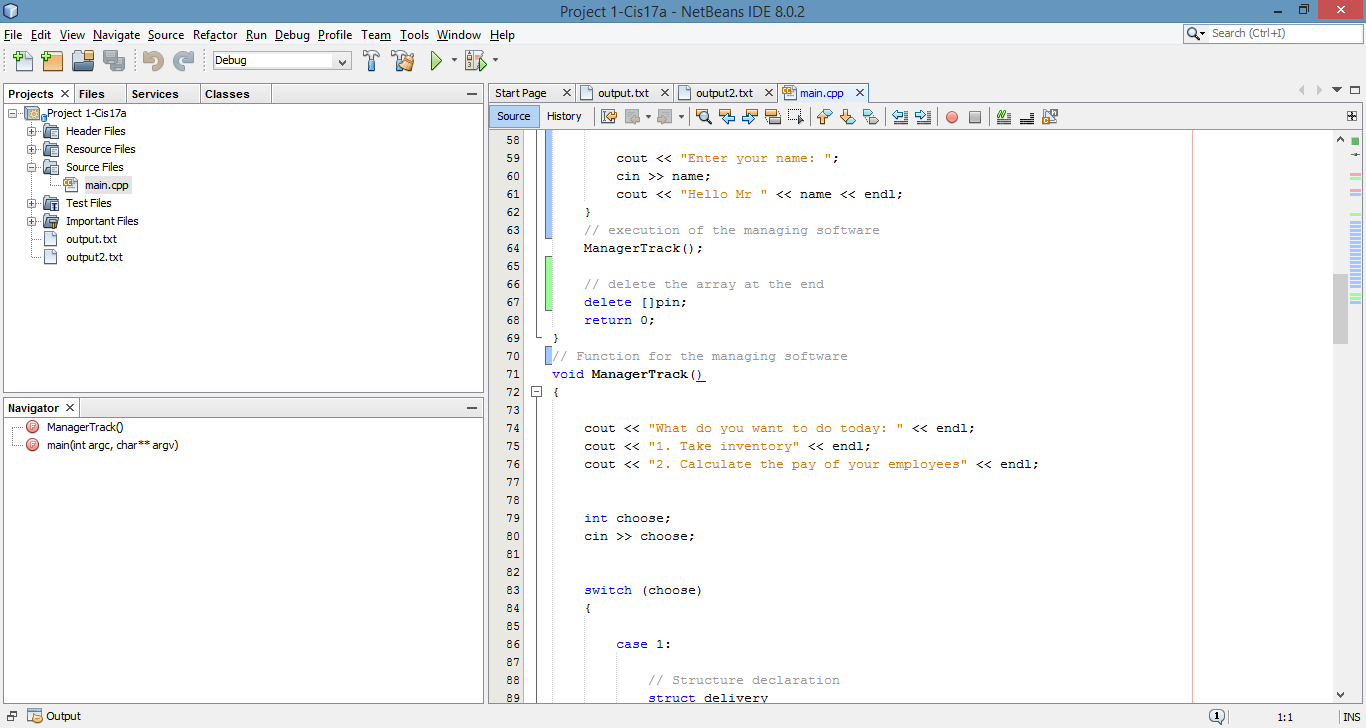
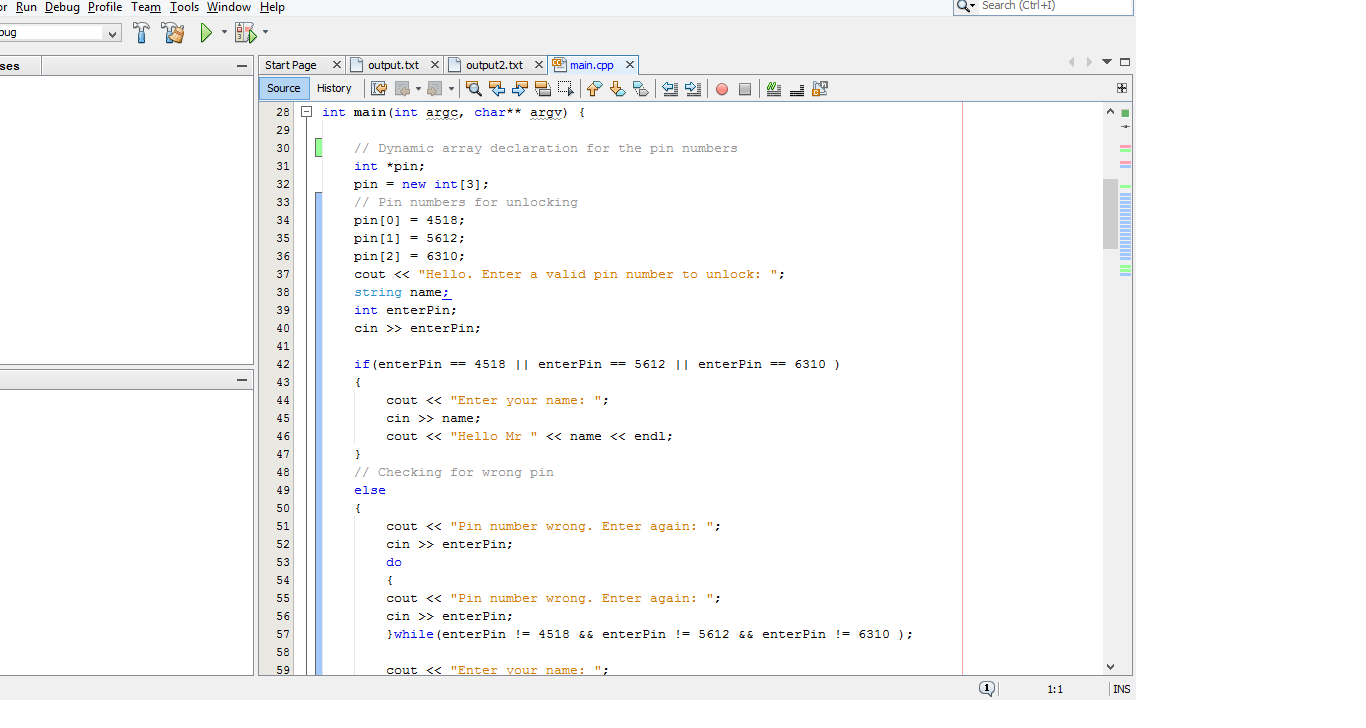
The main goal of the program is to mainly keep track of all the entries by writing the to a files and keep appending new data every time more items comes in.

**Description**

The program is first protected so at the beginning a pin number is required and a log sheet is created, so it keep track of all log in to use the program. This program is mainly to facilitate record keeping.

**Uncomplete**

I couldn’t finish flowcharting the program, I just did the main menu flowchart.

This is the main program:

**Program Code**

**#include <cstdlib>**

**#include <iostream>**

**#include <iomanip>**

**#include <fstream>**

**using namespace std;**

**// Useful variables**

**// Constant variables**

**// Functions prototypes for the program**

**void ManagerTrack();**

**/\***

**\***

**\*/**

**// Beginning of the execution**

**int main(int argc, char\*\* argv) {**

**// Dynamic array declaration for the pin numbers**

**int \*pin;**

**pin = new int[3];**

**// Pin numbers for unlocking**

**pin[0] = 4518;**

**pin[1] = 5612;**

**pin[2] = 6310;**

**cout << "Hello. Enter a valid pin number to unlock: ";**

**string name;**

**int enterPin;**

**cin >> enterPin;**

**if(enterPin == 4518 || enterPin == 5612 || enterPin == 6310 )**

**{**

**cout << "Enter your name: ";**

**cin >> name;**

**cout << "Hello Mr " << name << endl;**

**}**

**// Checking for wrong pin**

**else**

**{**

**cout << "Pin number wrong. Enter again: ";**

**cin >> enterPin;**

**do**

**{**

**cout << "Pin number wrong. Enter again: ";**

**cin >> enterPin;**

**}while(enterPin != 4518 && enterPin != 5612 && enterPin != 6310 );**

**cout << "Enter your name: ";**

**cin >> name;**

**cout << "Hello Mr " << name << endl;**

**}**

**// execution of the managing software**

**ManagerTrack();**

**// delete the array at the end**

**delete []pin;**

**return 0;**

**}**

**// Function for the managing software**

**void ManagerTrack()**

**{**

**cout << "What do you want to do today: " << endl;**

**cout << "1. Take inventory" << endl;**

**cout << "2. Calculate the pay of your employees" << endl;**

**int choose;**

**cin >> choose;**

**switch (choose)**

**{**

**case 1:**

**// Structure declaration**

**struct delivery**

**{**

**string items; // Items delivered**

**int numberItem; // Amount delivered**

**};**

**int day;**

**int month;**

**int year;**

**// Date of delivery**

**cout << "Enter The date: ";**

**cin >> day >> month >> year;**

**cout << "How many categories of items where delivered: ";**

**int itemNum;**

**cin >> itemNum;**

**// Check for negative entry**

**if(itemNum < 0)**

**{**

**cout << "Invalid. Enter again: ";**

**cin >> itemNum;**

**do**

**{**

**cout << "Invalid. Enter again: ";**

**cin >> itemNum;**

**}while(itemNum < 0);**

**}**

**else**

**{**

**delivery received[itemNum];**

**string category;**

**int numReceived;**

**// For example beer, candy, soda.**

**cout << "Enter the different categories of item received" << endl;**

**for(int i = 0; i < itemNum; i++)**

**{**

**cout << "Category: ";**

**cin >> category;**

**received[i].items = category;**

**// Number of each item received**

**cout << "Number received: ";**

**cin >> numReceived;**

**received[i].numberItem = numReceived;**

**}**

**cout << "Delivery received on " << day << "/" << month << "/";**

**cout << year << endl;**

**for(int i = 0; i < itemNum; i++)**

**{**

**cout << left << setw(8) << received[i].items;**

**cout << left << setw(8) << received[i].numberItem;**

**cout << endl;**

**}**

**ofstream outfile1("output2.txt", ios::app);**

**outfile1 << fixed << showpoint << setprecision(2);**

**outfile1 <<left<< setw(16) << "Category" << setw(8) << "Amount" << endl;**

**for (int i = 0; i < itemNum; i++)**

**{**

**outfile1 << left << setw(16) << received[i].items;**

**outfile1 << left << setw(8) << received[i].numberItem;**

**}**

**// Close the file**

**outfile1.close();**

**}**

**break;**

**case 2:**

**// declaration of the structure**

**struct globalPayroll**

**{**

**string name; // name of the employees**

**int hours; // hours worked**

**float ratePay; // pay rate of the employee**

**float money; // Payment of the employee**

**};**

**cout << "How many employees are there in the company: " ;**

**int empl;**

**cin >> empl; // number of employee**

**int count = 1;**

**globalPayroll employees[empl]; // array of structure for employees**

**for(int i = 0; i < empl; i++)**

**{**

**cout << "Employee #" << count << ": ";**

**cin >> employees[i].name; // employee name**

**cout << "Number of hours worked: ";**

**cin >> employees[i].hours; // hours worked**

**// error checking for negative hours**

**while(employees[i].hours < 0)**

**{**

**cout << "Invalid hour. Enter again: ";**

**cin >> employees[i].hours;**

**}**

**cout << "Rate of pay: ";**

**cin >> employees[i].ratePay; // pay rate for each employee**

**// error checking for negative pay rate**

**while(employees[i].ratePay < 0)**

**{**

**cout << "Invalid hour. Enter again: ";**

**cin >> employees[i].ratePay;**

**}**

**count++;**

**}**

**//Calculate and display the pay of each employee**

**int count2 = 1;**

**cout << fixed << showpoint << setprecision(2);**

**for(int i = 0; i < empl; i++)**

**{**

**float money;**

**money = employees[i].hours \* employees[i].ratePay;**

**employees[i].money = money;**

**// output gross pay**

**cout << "Employee #" << count2 << ": ";**

**cout << employees[i].name << "'s Pay is $";**

**cout << money << endl;**

**count2++;**

**}**

**//Open a file for the output**

**ofstream outfile("output.txt", ios::app);**

**outfile << fixed << showpoint << setprecision(2);**

**outfile <<left<< setw(16) << "Employee Name" << setw(8) << "Payment" << endl;**

**for (int i = 0; i < empl; i++)**

**{**

**// Writing the data to the file**

**outfile << left << setw(16) << employees[i].name ;**

**outfile << setw(8) << employees[i].money;**

**cout << endl;**

**}**

**// Close the file**

**outfile.close();**

**break;**

**}**

**}**