# p7.10

Henry Hsu

October 28, 2011

## 1 Specification

Complete the copier simulation of How To 7.1 on page 314. Your program should first show the main menu:

U)ser A)dministrator Q)uit

For a user, prompt for the ID and the number of copies, increment the appropriate account, and return to the main menu.

For an administrator, show this menu:

B)alance M)aster P)roject

In the balance option, show the balances of the master account and the ten project accounts. In the master option, prompt for a user ID and link it to the master account. In the project option, prompt for user and project IDs. Afterward, return to the main menu.

## 2 Analysis/Design

- Program main menu solicits user for input
- If input is U, run User Menu function. This function asks user for ID and copy amount. Increment the ID by copy amount
- If input is A, run Admin Menu function. This will allow the user to choose from B)Alance, M)aster, or P)roject submenus
- If input B from admin menu, display balance info on master account and 10 project accounts
- If input M from admin menu, ask user for an ID to link to the master account
- If input P from admin menu, ask user to link a user ID to a project account
- if input is Q, exit the program

# 3 Implementation

"p7_10.cpp" 3≡

⟨ *include files* 9 ⟩
⟨ *main menu* 4 ⟩
⟨ *user menu* 5 ⟩
⟨ *admin menu* 6a ⟩
⟨ *admin balance* 6b ⟩
⟨ *admin master* 7 ⟩
⟨ *admin project* 8 ⟩

```
int main()
{
        int master_account = 0;
        int project_accounts [10] = {0};        // init project id array

        int* users[101];
        for (int i = 0; i < 101; i++)           // Links all users to master account
        {
                users[i] = &master_account;
        }

        for (int i = 0; i < 10; i++)            // Links sample ID to  accounts
        {
                users[i] = &project_accounts[i];
        }

        string menu_option, admin_option;
        while (menu_option != "Q")
        {
                menu_option = main_menu();

                if (menu_option == "U")
                {
                        user_menu(users, &master_account);
                }

                else if (menu_option == "A")
                {
                        admin_option = admin_main_menu();
                        if (admin_option == "B")
                        {
                                admin_balance (&master_account, project_accounts);
                        }
                        else if (admin_option == "M")
                        {
                                int user_master = new_master ();
                                users[user_master] = &master_account;
                        }
                        else if (admin_option == "P")
                        {
                                project_menu (users, project_accounts);

                        }
                }
        }
        cout << endl;
        cout << "Thank you for using Copier Project v1.0." << endl;
}
```

⟨ *main menu* 4 ⟩ ≡

```
string main_menu()                              // Main program menu
{
        string menu_option;
        cout << "U)ser   A)dministrator   Q)uit: ";
        cin >> menu_option;
        cout << endl;
        menu_option[0] = toupper(menu_option [0]);      // converts 1st letter to uppercase
        menu_option.resize(1);                  // resizes input to one character as backup
        return menu_option;                     // returns user input
}
```

◇

Fragment referenced in 3.

⟨ *user menu* 5 ⟩ ≡

```
void user_menu(int* users[], int* master_account)       // User menu
{
        bool continue_input = true;
        int user_id = 0, copy_amount = 0;

        do
        {
                cout << "Please enter your user ID: ";
                cin >> user_id;
                if (cin.fail() || user_id < 1 || user_id > 100)
                {
                        cout << "Invalid ID. Please re-enter a valid ID. " << endl;
                        cin.clear();
                        cin.ignore(1);
                }
                else
                {
                        cout << "Number of copies: ";
                        cin >> copy_amount;
                        if (cin.fail ())
                        {
                                cout << "Invalid input. Exiting program";
                                exit (1);                       // critical error
                        }
                        else
                        {
                                *users[user_id] = *users[user_id] + copy_amount;
                                cout << endl << "Master Account Balance: " << *master_account <<
                                continue_input = false;         // returns to main menu
                        }
                }
        }
        while (continue_input);
}
```

◇

Fragment referenced in 3.

⟨ *admin menu* 6a ⟩ ≡

```
string admin_main_menu ()                       // Admin main menu
{
        string menu_option;
        cout << "B)alance   M)aster   P)roject: ";
        cin >> menu_option;
        cout << endl;
        menu_option[0] = toupper(menu_option [0]);      // converts 1st letter to uppercase
        menu_option.resize(1);                  // resizes input to one character as backup
        cout << endl;
        return menu_option;                     // returns user input
}
```

◇

Fragment referenced in 3.

⟨ *admin balance* 6b ⟩ ≡

```
// Displays the account balance
void admin_balance(int* master_account, int project_accounts[])
{
        cout << endl << "Master account Balance: " << *master_account << endl << endl;

        cout << "Project Accounts:" << endl << endl;
        for (int i = 0; i < 10; i++)
        {
                cout << "Account " << i << "'s balance is: ";
                cout <<  project_accounts[i] << endl;
        }
        cout << endl;
}
```

◇

Fragment referenced in 3.

⟨ *admin master* 7 ⟩ ≡

```
// Links a user account to the master account
int new_master()
{
        int new_master = 0;
        bool continue_input = true;
        do
        {
                cout << "Please enter a user ID to link it to the master account: ";
                cin >> new_master;
                if (new_master < 1 || new_master > 100)
                {
                        cout << "Valid user ID is from the range of 1 - 100." << endl;
                }
                else
                {
                        cout << "User ID " << new_master
                        << " is now linked to the master account." << endl;
                        continue_input = false;
                }
        }
        while (continue_input);
        return new_master;
}
```

◇

Fragment referenced in 3.

⟨ *admin project* 8 ⟩ ≡

```
int project_menu (int* users[], int project_accounts[])
{
        int user_id = 0, project_id = 0;
        bool continue_input = true;

        do
        {
        cout << "Please enter your user ID: ";
        cin >> user_id;
        if (user_id < 1 || user_id > 100)
        {
                cout << "Valid user ID is from the range of 1 - 100." << endl;
        }
        else
        {
                cout << "Please enter your project ID: ";
                cin >> project_id;
                if (project_id < 0 || project_id > 9)
                {
                        cout << "Valid project ID is from the range of 0 - 9." << endl;
                }
                else
                {
                        users[user_id] = &project_accounts[project_id];
                        cout << "User ID: " << user_id
                        << " is now linked to Project Account: " << project_id << "." << endl;
                        continue_input = false;
                }
        }

        }
        while (continue_input);


}
```

◇

Fragment referenced in 3.

These are the include files needed for library function calls

⟨ *include files* 9 ⟩ ≡

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;
```

◇

Fragment referenced in 3.

# 4 Test

```
C:\Users\Echo\Desktop\cs102>a
U)ser   A)dministrator   Q)uit: u 10 500 a b u 20 1000 a b a p 4 5 u 4 100 a b q


Please enter your user ID: Number of copies:
Master Account Balance: 500
```

```
Master account Balance: 1500

Project Accounts:

Account 0's balance is: 0
Account 1's balance is: 0
Account 2's balance is: 0
Account 3's balance is: 0
Account 4's balance is: 0
Account 5's balance is: 100
Account 6's balance is: 0
Account 7's balance is: 0
Account 8's balance is: 0
Account 9's balance is: 0

U)ser   A)dministrator   Q)uit:

Thank you for using Copier Project v1.0.
```