DEVELOPERS
All three members of the C++ Flashback team worked in almost equal amount for this homework.
   Jonathan Carver [signed]
   Darin Patch [signed]
   Joe Wiggins [signed]

DESIGN OVERVIEW
This will be the first version (ver. 2.0) of University People Management Software (UPMS) implemented by the team C++ Flashback. The prior versions were made individually by the members. This version will integrate from the previous versions some of the interface and functionality of the Student Record Management System. It will also include the capability of adding the following entities into the system:
   • Undergraduate students
   • Graduate students
   • Faculty
   • Staff

Version 2.0 will
1. Provide an interface to add a new instance of the entity with pertinent information for the entity through a menu driven interface program.
2. Provide an interface to update appropriate attributes for the entity
3. Allow the user to search the entire database for records with specific properties; and the system will include all the reasonable searchable attributes. Such as:
   • search a person by name
   • search a person by a university ID or social security number
   • search a person by name and date of birth
   • search all people by a zip code or a state name
   • search a person by major or department
   • search for the list of students in GPA order within a major/department possibly with a given range such as m <= gpa <= n.
   • search a person by an exact phone number or all people given an area code or an area code and a prefix
   • search for the list of all the courses a student has taken or is taking given the semester information
   • search for the list of all the students taking or has taken a specific course
   • search for the list of all the courses a faculty is teaching
   • search for the list of all the students a faculty is advising
   • search for the list of committee members given a graduate student information
   • list the employees supervised by an employee

PLATFORM
This program implemented in Microsoft Visual Studio C++ ver. 6.0
DESIGN DETAILS

For version 2.0 of the UPMS there will be 4 entities possible for entering into the system. They are: Undergraduate students, Graduate students, Faculty, and Staff.

The Undergraduate Student will have the following attributes:

- Name
- Address
- Phone number
- Date of birth
- ID number
- Major or pre-major
- List of courses taken and/or taking
- Email address
- Year (freshman, etc.)

The Graduate Student will have the following attributes:

- Name
- Address
- Phone number
- Date of birth
- ID number
- Major
- List of courses taken and/or taking
- Email address
- Three thesis committee members
- A thesis advisor
- Undergraduate major
- Undergraduate school
- Financial assistance status

The Faculty entity will have the following attributes:

- Name
- Address
- Phone number
- Date of birth
- ID number
- Department
- List of courses he/she is teaching in the current semester
- Email address
- A list of research interests
- A list of advisees
- Medical insurance plan
- Life insurance plan
- Retirement plan
- Annual salary

The Staff entity will have the following attributes:
- Name
- Address
- Phone number
- Date of birth
- ID number
- Department
- Email address
- Medical insurance plan
- Life insurance plan
- Retirement plan
- Annual salary
- Supervisor

```
PERSON
String Name
String Mname
String Lname
String Area
String Prefix
String Number
String Address
String Street
String Street2
String City
String State
String Zip
String DOB
String IDSSN
String Type

Insert_person (person): bool
Delete_person (person): bool
Clear (): void
read_from_file (ifstream): void
write_to_file (ofstream): void

Student
String Major
Vector Courses
String Email
String Status
String Id

Insert_student (student): bool
Delete_student (student): bool
Create_Id (): void
Clear (): void
read_from_file (ifstream): void
write_to_file (ofstream): void
```
### Graduate

- String major
- Vector course
- String email
- Vector thesis_mem
- String thesis_advisor
- String undergrad_major
- String undergrad_college
- String finance_status
- String id

- Insert_grad(graduate): bool
- Delete_grad(graduate): bool
- Create_Id(): void
- Clear(): void
- Read_from_file(ifstream): void
- Write_to_fileofstream): void

### Faculty

- String department
- Vector course
- String email
- Vector advisee
- Vector research
- String med_insurance
- String life_insurance
- String retirement
- String salary
- String id

- Insert_faculty(faculty): bool
- Delete_faculty(faculty): bool
- Create_Id(): void
- Clear(): void
- Read_from_file(ifstream): void
- Write_to_file ofstream): void

### Staff

- String department
- String email
- String med_insurance
- String life_insurance
- String retirement
- String salary
- String supervisor
- String id

- Insert_staff(staff): bool
- Delete_staff(staff): bool
- Read_from_file(ifstream): void
- Create_Id(): void
- Clear(): void
- Write_to_file ofstream): void
CHANGED TEMPLATES
None

NEW CLASSES
- Person
- Student
- Graduate
- Faculty
- Staff

DEPENDENCIES
The classes Student, Graduate, Faculty, and Staff will be dependent on Person. The classes will inherit the name, address, phone number, date of birth, and ID number from Person.

OPEN QUESTIONS

SIZING (IN PERSON DAYS)
7 Man-days

TARGET COMPLETION DATE
- Design Documentation: October 13, 2001
- Implementation: October 18, 2001
// Person.h: interface for the Person class.

#include "stdafx.h"
#include <string>
#include <iostream>
#include <fstream>

using namespace std;

class Person
{
public:
    void write_to_file(ofstream &file);
    void read_from_file(ifstream &file);
    void clear();
    bool delete_person(const Person &who);
    bool insert_person(const Person &who);
    Person();
    virtual ~Person();

protected:
    string fname;
    string mname;
    string lname;
    string area;
string prefix;
string number;
string address;
string street;
string street2;
string city;
string state;
string zip;
string dob;
string ssn;
string type;
}

#endif
!defined(AFX_PERSON_H__E6DBE987_C031_11D5_882A_525405F3FD9A__INCLUDED_)
// Student.h: interface for the Student class.

#include "Person.h"

class Student : public Person
{
public:
    Student();
    virtual ~Student();
    bool insert_student(const Student &who);
    bool delete_student(const Student &who);
    void create_id();
    void clear();
    void read_from_file(ifstream &file);
    void write_to_file(ofstream &file);

private:
    string major;
    vector<string> courses;
    string email;
    string status;
    string id;
};

#else
!defined(AFX_STUDENT_H__E6DBE988_C031_11D5_882A_525405F3FD9A__INCLUDED_)

#endif /
// Graduate.h: interface for the Graduate class.

//

#include "Person.h"

class Graduate : public Person
{

public:
    void write_to_file(ofstream &);  
    void read_from_file(ifstream &);  
    void clear();  
    void create_id();  
    bool delete_grad(const Graduate &who);  
    bool insert_grad(const Graduate &who);  
    Graduate();  
    virtual ~Graduate();

private:
    string major;  
    vector<string> course;  
    string email;  
    vector<string> thesis_mem;  
    string thesis_advisor;  

#if !_MSC_VER > 1000
    #pragma once
#endif // _MSC_VER > 1000

#define AFX_GRADUATE_H__E6DBE989_C031_11D5_882A_525405F3FD9A__INCLUDED_
string undergrad_major;
string undergrad_college;
string finance_status;
string id;

};

#ifdef AFX_GRADUATE_H__E6DBE989_C031_11D5_882A_525405F3FD9A__INCLUDED
#endif
// Faculty.h: interface for the Faculty class


#include "Person.h"

class Faculty : public Person
{
public:
    void write_to_file(ofstream &file);
    void read_from_file ifstream &file);
    void clear();
    void create_id();
    bool delete_faculty(const Faculty &who);    
    bool insert_faculty(const Faculty &who);    
    Faculty();
    virtual ~Faculty();

private:
    string department;
    string email;
    vector<string> course;
    vector<string> advisee;
    vector<string> research;
    string med_insurance;
string life_insurance;
string retirement;
string salary;
string id;

};

#undef //
defined(AFX_FACULTY_H__E6DBE98A_C031_11D5_882A_525405F3FD9AINCLUDED_)

// Staff.h: interface for the Staff class.

#include "Person.h"

class Staff : public Person
{
public:
    void write_to_file(ofstream &file);
    void read_from_file(ifstream &file);
    void clear();
    void create_id();
    bool delete_staff(const Staff &who);
    bool insert_staff(const Staff &who);
    Staff();
    virtual ~Staff();

private:
    string department;
    string email;
    string med_insurance;
    string life_insurance;
    string retirement;
    string salary;
    string supervisor;

#if !defined(AFX_STAFF_H__E6DBE98B_C031_11D5_882A_525405F3FD9A__INCLUDED_)
#define AFX_STAFF_H__E6DBE98B_C031_11D5_882A_525405F3FD9A__INCLUDED_
#endif

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
string id;

};

#endif //
!defined(AFX_STAFF_H__E6DBE98B_C031_11D5_882A_525405F3FD9A__INCLUDED_)