

Eduardo Martinez
Cs 211
2/24/16
assignment 2

car.h

```
#include <string>
using namespace std;
#ifndef CAR_H
#define CAR_H
class Car
{
private:
    string make;
    string model;
    int year;
    string color;
    int miles;
    int x;
    int y;

public:
    Car(string ,string ,int ,string ,int ,int ,int );

    void goForward(int num);
    void goBackward(int num);
    void goRight(int num);
    void goLeft(int num);
    void repaint(string cl);
    string getMake();
    string getModel();
    int getYear();
    int getMiles();
    string getColor();
    int getX();
    int getY();

};
#endif
```

car.C

```
#include <iostream>
#include "car.h"
using namespace std;

Car::Car(string mk, string md, int yr, string c, int mi, int bx,int by)
{
    make = mk;
    model = md;
    year = yr;
    color = c;
    miles = mi;
    x = bx;
```

```
    y = by;
}
void Car::goForward(int num)
{
    y += num;
    miles += num;
}

void Car::goBackward(int num)
{
    y -= num;
    miles += num;
}
void Car::goRight(int num)
{
    x += num;
    miles += num;
}
void Car::goLeft(int num)
{
    x -= num;
    miles += num;
}
void Car::repaint(string cl)
{
    color = cl;
}
string Car::getMake()
{
    return make;
}
string Car::getModel()
{
    return model;
}
int Car::getYear()
{
    return year;
}

int Car::getMiles()
{
    return miles;
}
string Car::getColor()
{
    return color;
}
int Car::getX()
{
    return x;
}
int Car::getY()
{
    return y;
}
```

```
}
```

carClient.C

```
#include <iostream>
#include "car.h"

using namespace std;

void viewCar(Car& c);
int main()
{
    Car car1("Toyota", "Celica", 2016, "white", 0 ,0, 0);
    Car car2("Honda", "Civic", 2008, "black", 20000, 2, 2);

    car1.goForward(3);
    car2.goBackward(1);
    car1.goRight(2);
    car2.goLeft(5);
    viewCar(car1);
    car1.repaint("red");
    viewCar(car1);
    viewCar(car2);

    return 0;
}

void viewCar(Car& c)
{
    cout << c.getColor() << " " << c.getMake() << " " << c.getModel() << " "
    << "(" << c.getYear() << ")" << " is at location (" << c.getX() << "," <<
    c.getY() << ")" << " with mileage " << c.getMiles() << " miles" << endl;
```

Car Program Test Run:

```
[marti540@empress cs211]$ g++ carClient.C car.C
[marti540@empress cs211]$ ./a.out
white Toyota Celica (2016) is at location (2,3) with mileage 5 miles
red Toyota Celica (2016) is at location (2,3) with mileage 5 miles
black Honda Civic (2008) is at location (-3,1) with mileage 20006 miles
[marti540@empress cs211]$
```

price.h

```
#ifndef PRICE_H
#define PRICE_H
```

```

class Price
{
private:
    int dollar;
    int cent;
public:
    Price();
    Price (int d, int c);

    Price operator+(const Price another);
    int getDollar();
    int getCent();
    Price operator-(const Price other);
};
#endif

```

price.C

```

#include "price.h"

Price::Price()
{
    dollar= 0;
    cent = 0;
}
Price::Price (int d, int c)
{
    dollar =d;
    cent =c;
}
Price Price::operator+(const Price another)
{
    dollar  = this->dollar + another.dollar;
    cent = this->cent + another.cent;
    if(cent >= 100)
    {
        dollar += (cent/100);
        cent = (cent%100);
    }
    Price c(dollar, cent);

    return c;
}
Price Price::operator-(const Price other)
{
    //convert each price to cents
    int p1 = this->dollar * 100 + this->cent;
    int p2 = other.dollar * 100 + other.cent;
    int diff;
    if(p1 >= p2)
    {
        diff = p1 - p2;
        Price diff0b(diff/100, diff%100);
    }
}

```

```

        return diff0b;
    }
    else
    {
        Price neg0b(-9999999, -9999999); //Don't do the math if p2 is less than
p1
        return neg0b;
    }
}
int Price::getDollar()
{
    return dollar;
}

int Price::getCent()
{
    return cent;
}

```

priceClient.C

```

#include <iostream>
#include "price.h"
using namespace std;

int main()
{
    Price p1(2, 50);
    Price p2(3, 75);
    Price p3 = p1 + p2;

    cout << "The total price is" << p3.getDollar() << " dollars and " <<
p3.getCent() << " cents" << endl; //The total price is 6 dollars and 25
cents
    return 0;
}

```

Price Program Test Run:

```

[marti540@empress Stu]$ g++ priceClient.C price.C
[marti540@empress Stu]$ ./a.out
The total price is6 dollars and 25 cents
[marti540@empress Stu]$

```

student.h

```

#ifndef STUDENT_H
#define STUDENT_H

#include <string>
#include <vector>
using namespace std;
#include "price.h"

class Student
{
private:
    static int id;
    static int numStu;
    int stId;
    string fName;
    string lName;
    char gender;
    Price balance;
    vector<int> coVec;

public:
    Student();
    Student(string f, string l, char g);
    bool addCourse(int crn);
    bool dropCourse(int crn);
    string getFirstName();
    string getLastName();
    char getGender();
    int getId();
    static int getNumStu(); //This function returns the static data member
    int getDollar();
    int getCent();
    vector<int> getCourses();
    void chargeFee(int d, int c);
    void reduceFee(int d, int c);
};

#endif

```

student.c

```

#include "student.h"

int Student::id =100; //initialize id to 100
int Student::numStu = 0; //initialize numStu to 0

Student::Student()
{
    stId = id;
    fName = "unknown";
    lName = "unknown";
    gender = 'X';
}

```

```

}
Student::Student(string f, string l, char g)
{
    numStu++; //one more student. increment numStu
    stId= id; //set stId of the new student to the value of the static member
    "id"
    fName = f; //initialize the data members
    lName = l;
    gender = g;
    id++; //increment id for stId of the next student
}

//returns true if crn is added to the vector
//returns false if crn already exists. Do not add the same crn into the
vector
bool Student::addCourse(int crn)
{
    //If crn already exists in the vector, don't add it
    for(int i = 0; i < coVec.size(); i++)
        if(crn == coVec[i])
            return false;

    //If the vector doesn't have the same crn, add it to the vector
    coVec.push_back(crn);
    return true;
}

//return true if the crn is dropped from vector
//return false if the crn is not found in vector
bool Student::dropCourse(int crn)
{
    for(int i = 0; i < coVec.size(); i++)
    {
        if(crn == coVec[i])
        {
            coVec.erase(coVec.begin()+i);
            return true;
        }
    }

    return false;
}

string Student::getFirstName()
{
    return fName;
}

string Student::getLastName()
{
    return lName;
}

char Student::getGender()

```

```

{
    return gender;
}

int Student::getId()
{
    return stId;
}

int Student::getNumStu()
{
    return numStu;
}

int Student::getDollar()
{
    return balance.getDollar();
}

int Student::getCent()
{
    return balance.getCent();
}

//Returns the vector of course the student is enrolled in
vector<int> Student::getCourses()
{
    return coVec;
}

//$d.c is added to the student's fee
void Student::chargeFee(int d, int c)
{
    Price p(d,c);

    balance = balance+ p;
}

//The student's fee is reduced by $d.c
void Student::reduceFee(int d, int c)
{
    Price p(d,c);

    balance = balance - p;
}

```

studentCleint.C

```

#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector> //include vector.h
using namespace std;
#include "student.h"
#include "inputCheck.h"

```



```

void showMenu();
void addStudentsFromFile(vector<Student>& s1);
void addOneStudent(vector<Student>& s2);
void printAllStudents(vector<Student>& s3);
void addCourseToStudent(vector<Student>& s4);
void dropCourseFromStudent(vector<Student>& s5);

int main()
{
    cout << "\n *** TEST: the number of students at the beginning is " <<
Student::getNumStu() << endl;

    vector<Student> s; //declare a vector of students
    int choice;
    do
    {

        showMenu();
        cout << "Enter your choice ----> ";
        choice = getNumberInRange(1, 6, "Invalid choice. Enter 1 through 6: ");

        switch(choice)
        {
            case 1:
                addStudentsFromFile(s);
                break;
            case 2:
                addOneStudent(s);
                break;
            case 3:
                printAllStudents(s);
                cout << "\n *** TEST: the number of students is " <<
Student::getNumStu() << endl;
                break;
            case 4:
                addCourseToStudent(s);
                break;
            case 5:
                dropCourseFromStudent(s);
                break;
            case 6:
                cout << "Ending the program ....." << endl;
                break;
        }
    }while(choice != 6);

    cout << "\n *** TEST: the number of students at the end is " <<
Student::getNumStu() << endl;

    return 0;
}

void showMenu()
{

```

```

    cout <<
"\n*****" <<
endl;
    cout << "1: Add students from a file" << endl;
    cout << "2: Add one student" << endl;
    cout << "3: Print all students" << endl;
    cout << "4: Add a course to a student" << endl;
    cout << "5: drop a course from a student" << endl;
    cout << "6: End the program" << endl;
}

//This function adds students from an input file
//e.g. for input file (for each student, first last gen)
// Mike Smith M
// Kathy Ross F
void addStudentsFromFile(vector<Student>& s1)
{
    string fileName;

    cout << "Enter the file name: ";
    cin >> fileName;

    ifstream fin;
    fin.open(fileName.c_str()); //open the file

    if(!fin)
    {
        cout << fileName << " doesn't exist " << endl;
    }
    else//The file exists
    {
        string fn, ln;
        char gn;

        fin >> fn >> ln >> gn;//read the first student
        while(fin)//if reading was successful, enter the while
        {
            Student ob(fn,ln,gn);//create a new student with the data from
input
            s1.push_back(ob); //add it to the vector
            fin >> fn >> ln >> gn; //read the next student
        }
    }
}

//This function adds one student to the vector.
void addOneStudent(vector<Student>& s2)
{
    string fn, ln;
    char gn;

    cout << "Enter the student's first name: ";
    cin >>fn;
    cout << "Enter the student's last name: ";
    cin >>ln;

```

```

        cout << "Enter the student's gender: ";
        gn = getGender("Invalid input. Enter M or F: ");

        Student ob1(fn,ln,gn); //create a new student with the data from the
user's input
        s2.push_back(ob1); //add the new student to the vector
    }

//Print the information about all the students
void printAllStudents(vector<Student>& s3)
{
    //Show the labels
    cout << left << setw(10) << "Id" << setw(10) << "First" << setw(10) <<
"Last" << right << setw(10) << "Fee" << left << setw(10) << " Courses" <<
endl;

    //Go through each student in the vector
    for(int n = 0; n < s3.size() ; n++)
    {
        cout << setw(10) << s3[n].getId() << setw(10) <<
s3[n].getFirstName() << setw(10) << s3[n].getLastName() << right << setw(7)
<< s3[n].getDollar() << ".";

        //If the cent is less than 10, show a 0 first.
        //e.g. cent = 3, 03 will be shown.
        if(s3[n].getCent() < 10)
            cout << "0";
        cout << s3[n].getCent();

        //show all courses the student is enrolled in
        cout << left << " ";
        vector<int> co = s3[n].getCourses(); //get the vector of courses for
this student
        for(int c = 0; c < co.size(); c++)
            cout << setw(6) << co[c];

        cout << endl;
    }
}

//Add a new course to a student.
//Charge $50.10 per course
void addCourseToStudent(vector<Student>& s4)
{
    bool fnd=false;
    bool exist;
    int num;

    cout << "Enter the student id: ";
    int id = getNumberInRange(1, 99999, "Invalid id. Enter the student id:
");
    cout << "Enter the CRN: ";
    int crn = getNumberInRange(1, 99999, "Invalid CRN. Enter the CRN: ");

```

```

for(int i=0;i < s4.size(); i++)
{
    if(s4[i].getId() == id)
    {
        fnd=true;
        num= i;
    }
}
if(fnd)
{
    exist = s4[num].addCourse(crn);
    if(exist)
    {
        s4[num].chargeFee(50,10);
        cout << crn << " was added" << endl;
    }
    else
        cout << "ERROR: The student is already enrolled in " << crn << endl;
}
else
    cout<< "ERROR: The student wasn't found" << endl;
}

//drop a course from a student
//reduct $50.10 from the student's balance
void dropCourseFromStudent(vector<Student>& s5)
{
    bool fnd = false;
    bool exist;
    int num;

    cout << "Enter the student id: ";
    int id = getNumberInRange(1, 99999, "Invalid id. Enter the student id:
");
    cout << "Enter the CRN: ";
    int crn = getNumberInRange(1, 99999, "Invalid CRN. Enter the CRN: ");

    for(int i=0;i < s5.size(); i++)
    {
        if(s5[i].getId() == id)
        {
            fnd=true;
            num= i;
        }
    }
    if(fnd)
    {
        exist =s5[num].dropCourse(crn);
        if(exist)
        {
            s5[num].reduceFee(50,10);
            cout << crn << " was dropped" << endl;

```

```

    }
    else
        cout <<"ERROR: The student is not enrolled in " <<crn << endl;
    }
    else
        cout<< "ERROR: The student wasn't found" << endl;
}

```

Student Program Test Run:

```

[marti540@empress Stu]$ g++ studentClient.C student.C price.C
[marti540@empress Stu]$ ./a.out

```

*** TEST: the number of students at the beginning is 0

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 1
Enter the file name: stu1.dat

```

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 3

```

Id	First	Last	Fee	Courses
100	Mike	Smith	0.00	
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	

*** TEST: the number of students is 5

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 2
Enter the student's first name: Kaz
Enter the student's last name: Slo
Enter the student's gender: 100

```

100 is invalid. Invalid input. Enter M or F: x
 x is invalid. Invalid input. Enter M or F: F

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program

Enter your choice ----> 3

Id	First	Last	Fee	Courses
100	Mike	Smith	0.00	
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	
105	Kaz	Slo	0.00	

*** TEST: the number of students is 6

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program

Enter your choice ----> 1

Enter the file name: stu2.dat

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program

Enter your choice ----> 3

Id	First	Last	Fee	Courses
100	Mike	Smith	0.00	
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	
105	Kaz	Slo	0.00	
106	Rich	Watson	0.00	
107	Kit	Watkins	0.00	

*** TEST: the number of students is 8

1: Add students from a file
 2: Add one student
 3: Print all students

4: Add a course to a student
 5: drop a course from a student
 6: End the program
 Enter your choice ----> 4
 Enter the student id: mike
 Invalid id. Enter the student id: 100
 Enter the CRN: mike
 Invalid CRN. Enter the CRN: 1111
 11111 was added

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program
 Enter your choice ----> 3

Id	First	Last	Fee	Courses
100	Mike	Smith	50.10	11111
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	
105	Kaz	Slo	0.00	
106	Rich	Watson	0.00	
107	Kit	Watkins	0.00	

*** TEST: the number of students is 8

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program
 Enter your choice ----> 4
 Enter the student id: 443
 Enter the CRN: 234
 ERROR: The student wasn't found

1: Add students from a file
 2: Add one student
 3: Print all students
 4: Add a course to a student
 5: drop a course from a student
 6: End the program
 Enter your choice ----> 4
 Enter the student id: 100
 Enter the CRN: 1111
 ERROR: The student is already enrolled in 1111

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 4
Enter the student id: 100
Enter the CRN: 2222
2222 was added

```

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 3

```

Id	First	Last	Fee	Courses
100	Mike	Smith	100.20	11111 2222
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	
105	Kaz	Slo	0.00	
106	Rich	Watson	0.00	
107	Kit	Watkins	0.00	

*** TEST: the number of students is 8

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 5
Enter the student id: mike
Invalid id. Enter the student id: 100
Enter the CRN: mike
Invalid CRN. Enter the CRN: 11111
11111 was dropped

```

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 3

```

Id	First	Last	Fee	Courses
100	Mike	Smith	50.10	2222

101	Kathy	Ross	0.00
102	Tom	Cruse	0.00
103	Albert	Einstein	0.00
104	Ada	Augusta	0.00
105	Kaz	Slo	0.00
106	Rich	Watson	0.00
107	Kit	Watkins	0.00

*** TEST: the number of students is 8

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 5
Enter the student id: 888
Enter the CRN: 2222
ERROR: The student wasn't found

```

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 5
Enter the student id: 100
Enter the CRN: 2222
2222 was dropped

```

```

1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 3

```

Id	First	Last	Fee	Courses
100	Mike	Smith	0.00	
101	Kathy	Ross	0.00	
102	Tom	Cruse	0.00	
103	Albert	Einstein	0.00	
104	Ada	Augusta	0.00	
105	Kaz	Slo	0.00	
106	Rich	Watson	0.00	
107	Kit	Watkins	0.00	

*** TEST: the number of students is 8

```
1: Add students from a file
2: Add one student
3: Print all students
4: Add a course to a student
5: drop a course from a student
6: End the program
Enter your choice ----> 7
Invalid choice. Enter 1 through 6: 6
Ending the program .....
```

```
*** TEST: the number of students at the end is 8
[marti540@empress Stu]$
```