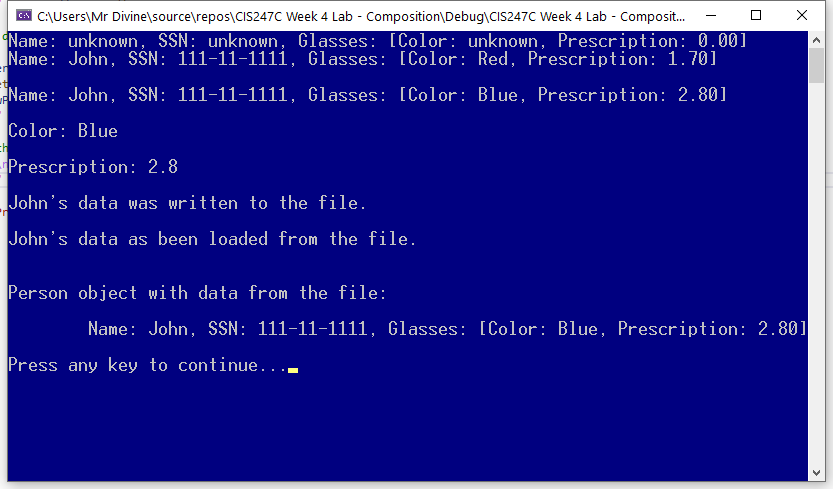
Screenshots:



**Code:**

# Source.cpp

/\*

Edward Alvarado

Class: CIS247C

Date: 3/24/2021

Week 4 Lab - Compositions

\*/

#include <iostream>

#include <string>

#include <conio.h>

#include "Person.h"

#include "Glasses.h"

using namespace std;

/// Entry point to the applicatrion

int main()

{

//creating a default person object

Person person1;

cout << person1.toString() << endl;

// change the values by setrting the nameand ssn, and add glasses object

person1.setName("John");

person1.setSsn("111-11-1111");

Glasses g1("Red", 1.70f);

person1.setMyGlasses(g1);

cout << person1.toString() << endl;

//change the glasses using an anonymous Glasses object

person1.setMyGlasses(Glasses("Blue", 2.8f));

cout << "\n" << person1.toString() << endl;

// show only color

cout << "\nColor: " << person1.getMyGlasses().getColor() << endl;

// show the prescription only

cout << "\nPrescription: " << person1.getMyGlasses().getPrescription() << endl;

// write the person object's state to the file

string result = person1.writeToFile();

cout << "\n" << result << endl;

// create a default object, change the name, and read the data into the new object

Person newPerson;

newPerson.setName("John");

result = newPerson.readFromFile();

cout << "\n" << result << endl;

// display the data that was read from the file

cout << "\n\nPerson object with data from the file:\n" << endl;

cout << "\t" << newPerson.toString() << endl;

// Pause

cout << "\nPress any key to continue...";

\_getch();

return 0;

}

# Person.h

#pragma once

#include <string>

#include <fstream> // required to read/write files

#include "Glasses.h"

using namespace std;

class Person

{

private:

//attributes

string name;

string ssn;

Glasses myGlasses; // Creates default glasses object

public:

// constructors and destructor

Person(void);

Person(string name, string ssn, Glasses theGlasses);

~Person(void);

//behaviors

string toString(void);

string writeToFile(void);

string readFromFile(void);

//accessors and mutators

string getName(void);

void setName(string name);

string getSsn(void);

void setSsn(string ssn);

Glasses getMyGlasses(void);

void setMyGlasses(Glasses theGlasses);

};

# Person.cpp

#include "Person.h"

Person::Person()

{

name = "unknown";

ssn = "unknown";

}

Person::Person(string name, string ssn, Glasses theGlasses)

{

setName(name);

setSsn(ssn);

setMyGlasses(theGlasses);

}

Person::~Person(void)

{

}

//behaviors

string Person::toString(void)

{

return "Name: " + name + ", SSN: " + ssn + ", Glasses: [" + myGlasses.toString() + "]";

}

string Person::writeToFile(void)

{

//write the state of the Person to the file using the person's name

string fileName = name + ".txt";

ofstream ofile(fileName); //defaults to ios::out (destroys prior data) -- ios::app --> appends to the file

ofile << name << endl;

ofile << ssn << endl;

ofile << myGlasses.getColor() << endl;

ofile << myGlasses.getPrescription() << endl;

ofile.close();

// tell user that data was written to the file

return name + "'s data was written to the file.";

}

string Person::readFromFile(void)

{

// read from the file

string fileName = name + ".txt";

ifstream ifile(fileName);

if (!ifile.is\_open()) //! mean not--> if (ifile.is\_open() == false);

{

return "Error. File cannot be opened. Does it exist?";

}

else

{

getline(ifile, name); // read into this object's name attribute

getline(ifile, ssn); // read into this objects's ssn attribute

string color = "";

getline(ifile, color); // read the glasses' color

float pers = 0.0f;

ifile >> pers; // read the glasses' prescription

this->setMyGlasses(Glasses(color, pers));

ifile.ignore(1); // ingore and use up the 'n' (newline)

ifile.close();

// tell the user data as been loaded

return name + "'s data as been loaded from the file.";

}

}

//accessors and mutators

string Person::getName(void)

{

return name;

}

void Person::setName(string name)

{

if (name.length() > 0)

this->name = name;

else

this->name = "unknown";

}

string Person::getSsn(void)

{

return ssn;

}

void Person::setSsn(string ssn)

{

if (ssn.length() == 9 || ssn.length()==11)

this->ssn = ssn;

else

this->ssn = "unknown";

}

Glasses Person::getMyGlasses(void)

{

return myGlasses;

}

void Person::setMyGlasses(Glasses theGlasses)

{

this->myGlasses = theGlasses;

}

# Glasses.h

#pragma once

#include <string>

using namespace std;

class Glasses

{

private:

// attributes

string color;

float prescription;

public:

// constructors and destructor

Glasses(void);

Glasses(string color, float prescription);

~Glasses(void);

//behaviors

string toString(void);

//accessors and mutators

string getColor(void);

void setColor(string color);

float getPrescription(void);

void setPrescription(float prescription);

};

# Glasses.cpp

#include "Glasses.h"

Glasses::Glasses()

{

color = "unknown";

prescription = 0.0f; // f will create a float instead of double

}

Glasses::Glasses(string color, float prescription)

{

setColor(color);

setPrescription(prescription);

}

Glasses::~Glasses(void)

{

}

//behaviors

string Glasses::toString(void)

{

string strPrescrption = to\_string(prescription);

strPrescrption = strPrescrption.substr(0, 4); // create a substring shortening the number of characters shown

return "Color: " + color + ", Prescription: " + strPrescrption;

}

//accessors and mutators

string Glasses::getColor(void)

{

return color;

}

void Glasses::setColor(string color)

{

if (color.length() > 0)

this->color = color;

else

this->color = "unknown";

}

float Glasses::getPrescription(void)

{

return prescription;

}

void Glasses::setPrescription(float prescription)

{

if (prescription > 0.0f)

this->prescription = prescription;

else

this->prescription = 0.0;

}