Program No. : Farmer\_Garcia\_Elliot\_H7.cpp

Programmer : Elliot Farmer Garcia

Course/Section : ELET 2300-07/23493

Instructor : Dr. F. Attarzadeh

Date Assigned : 11/12/2019

Date Modified : 11/17/2019

Due Date : 11/21/2019

Compiler : Microsoft Visual Studio Enterprise 2019

Environment : Console Applications

Operating System : Windows 7

**Program Flowchart**



**.cpp source file**

/\*

Program No. : Farmer\_Garcia\_Elliot\_H7.cpp

Programmer : Elliot Farmer Garcia

Course / Section : ELET 2300 - 07 / 23493

Instructor : Dr. F. Attarzadeh

Date Assigned : 11/12/2019

Date Modified : 11/17/2019

Due Date : 11/21/2019

Compiler : Microsoft Visual Studio Enterprise 2019

Environment : Console Applications

Operating System : Windows 7

\*/

/\*

Problem Statement

This menu-driven program allows the user to compare their desktop or laptop

computer to an Apple desktop or laptop computer. The manufacturer, model, and

year are compared for desktops, and the manufacturer, model, and weight are

compared for laptops.

Enter (d) or (D) to compare desktops, or (l) or (L) to compare laptops.

(h) or (H) prints the help menu. (q) or (Q) to quit.

Output is aligned to the default Windows 7 terminal size, 80x25.

\*/

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

//base class Computer declaration

class Computer {

string manufacturer;

string model;

public:

Computer(); //default constructor

Computer(string ma, string mo); //overloaded constructor

~Computer(); //destructor

void setManufacturer(string ma); //mutator

void setModel(string mo); //mutator

void displayComputer(void); //accessor

};

//derived class Desktop declaration, inherits Computer

class Desktop : public Computer {

int year;

public:

Desktop(); //default constructor

Desktop(string ma, string mo, int y); //overloaded constructor

~Desktop(); //destructor

void setYear(int y); //mutator

void displayDesktop(void); //accessor

};

//derived class Laptop declaration, inherits Computer

class Laptop : public Computer {

float weight;

public:

Laptop(); //default constructor

Laptop(string ma, string mo, float w); //overloaded constructor

~Laptop(); //destructor

void setWeight(float w); //mutator

void displayLaptop(void); //accessor

};

void help(void);

Desktop defaultDesktop(void);

Laptop defaultLaptop(void);

Desktop customDesktop(void);

Laptop customLaptop(void);

void compareDesktops(Desktop D1);

void compareLaptops(Laptop L1);

//main creates instances of Desktop and Laptop with default values, and

//handles menu operations. The default computers are then sent to either

//compareDesktops() and compareLaptops() upon menu request.

int main() {

//variable set by user to call the desired operation

char op;

//strings for menu handling

string header = "This program allows you to visually compare stats for "

"your computer against\nApple products.\n";

string menu = "\n\t\tHelp\t\tDesktop\t\tLaptop\t\tQuit\n\n";

string prompt = "Please enter the initial of your desired operation: ";

string inv = "Invalid selection.\n";

//default computers are created with default constructors

Desktop D1 = defaultDesktop();

Laptop L1 = defaultLaptop();

cout << header;

//loop runs until 'q' or 'Q' is selected

while (true) {

//prints menu and asks user for menu selection

cout << menu;

cout << prompt;

cin >> op;

cout << "\n";

//menu selection is evaluated

switch (op) {

//help has been selected

case 'h':

case 'H':

help();

break;

//desktop has been selected

case 'd':

case 'D':

compareDesktops(D1);

break;

//laptop has been selected

case 'l':

case 'L':

compareLaptops(L1);

break;

//quit program has been selected

case 'q':

case 'Q':

return 0; //exits program

//invalid menu selection, menu repeats

default:

cout << inv;

}

}

}

//default Computer constructor

Computer::Computer() {

manufacturer = "";

model = "";

}

//overloaded Computer constructor, sets manufacturer and model to parameters

Computer::Computer(string ma, string mo) {

manufacturer = ma;

model = mo;

}

//Computer destructor

Computer::~Computer() {

}

//mutator that sets computer manufacturer

void Computer::setManufacturer(string ma) {

manufacturer = ma;

}

//mutator that sets computer model

void Computer::setModel(string mo) {

model = mo;

}

//accessor function - prints manufacturer and model of Computer

void Computer::displayComputer() {

cout << manufacturer << " " << model;

}

//default Desktop constructor

Desktop::Desktop() : Computer() {

year = 0;

}

//overloaded Desktop constructor, allows user to set custom manufacturer,

//model, and year

Desktop::Desktop(string ma, string mo, int y) : Computer(ma, mo) {

year = y;

}

//Desktop destructor

Desktop::~Desktop() {

}

//mutator function, sets Desktop year

void Desktop::setYear(int y) {

year = y;

}

//accessor function - prints manufacturer, model, and year of Desktop

void Desktop::displayDesktop(void) {

//prints manufacturer and model using inherited displayComputer()

Computer::displayComputer();

//prints year

cout << " (" << year << " edition)\n";

}

//default Laptop constructor

Laptop::Laptop() : Computer() {

weight = 0.0f;

}

//overloaded Laptop constructor, allows user to set custom manufacturer,

//model, and weight

Laptop::Laptop(string ma, string mo, float w) : Computer(ma, mo) {

weight = w;

}

//Laptop destructor

Laptop::~Laptop() {

}

//Laptop mutator function, sets weight

void Laptop::setWeight(float w) {

weight = w;

}

//accessor function - prints Laptop manufacturer, model, and weight

void Laptop::displayLaptop(void) {

//prints manufacturer and model using inherited displayComputer()

Computer::displayComputer();

//prints weight

cout << " (" << fixed << setprecision(1) << weight << " lbs.)\n";

}

void help(void) {

string help = "This program helps the user visually compare their "

"computer to the latest\ncomputers by Apple.\n\n"

"Entering 'd' or 'D' will allow the user to compare desktops. The "

"user will\nsubsequently be asked to enter the manufacturer and "

"model name, as well as the\nmodel year (int).\n\n"

"Entering 'l' or 'L' will allow the user to compare laptops. The "

"user will\nsubsequently be asked to enter the manufacturer and "

"model name, as well as the\nmodel weight (float).\n\n"

"Entering 'q' or 'Q' will quit the program.\n\n";

cout << help;

}

//creates Desktop using default constructor, and assigns hardcoded values

//for the user to compare their Desktop against

Desktop defaultDesktop() {

//default Desktop constructor

Desktop D;

//set D to 2019 Apple Mac Pro using mutator functions

D.setManufacturer("Apple");

D.setModel("Mac Pro");

D.setYear(2019);

return D;

}

//creates Laptop using default constructor, and assigns hardcoded values

//for the user to compare their Laptop against

Laptop defaultLaptop() {

//default Laptop

Laptop L;

//set L to Apple MacBook Pro weighing 4.0lbs using mutator functions

L.setManufacturer("Apple");

L.setModel("MacBook Pro");

L.setWeight(4.0);

return L;

}

//prompts user for values to create a custom desktop

Desktop customDesktop(void) {

//values for custom Desktop

string ma, mo;

int y;

cin.ignore(); //clear input buffer

//user set custom Desktop values

cout << "Please enter desktop manufacturer: ";

getline(cin, ma); //include potential spaces

cout << "Please enter desktop model: ";

getline(cin, mo); //include potential spaces

cout << "Please enter desktop year: ";

cin >> y;

return Desktop(ma, mo, y);

}

//prompts user for values to create a custom laptop

Laptop customLaptop(void) {

string ma, mo;

float w;

cin.ignore(); //clear input buffer

//user set custom Laptop values

cout << "Please enter laptop manufacturer: ";

getline(cin, ma); //include potential spaces

cout << "Please enter laptop model: ";

getline(cin, mo); //include potential spaces

cout << "Please enter laptop weight: ";

cin >> w;

return Laptop(ma, mo, w);

}

//prints values for default and custom desktops, allowing user to visually

//compare them

void compareDesktops(Desktop D1) {

//custom Desktop D2 is created with overloaded constructor

Desktop D2 = customDesktop();

cout << "\n";

D1.displayDesktop();

cout << "vs.\n";

D2.displayDesktop();

cout << "\n";

system("pause"); //waits for the user to press a key before continuing

}

//prints values for default and custom desktops, allowing user to visually

//compare them

void compareLaptops(Laptop L1) {

//custom Laptop L2 is created with overloaded constructor

Laptop L2 = customLaptop();

cout << "\n";

L1.displayLaptop();

cout << "vs.\n";

L2.displayLaptop();

cout << "\n";

system("pause"); //waits for the user to press a key before continuing

}

**Sample Runs**

This program allows you to visually compare stats for your computer against

Apple products.

Help Desktop Laptop Quit

Please enter the initial of your desired operation: d

Please enter desktop manufacturer: Dell

Please enter desktop model: New Inspiron 24 5000 All-in-One

Please enter desktop year: 2019

Apple Mac Pro (2019 edition)

vs.

Dell New Inspiron 24 5000 All-in-One (2019 edition)

Press any key to continue . . .

This program allows you to visually compare stats for your computer against

Apple products.

Help Desktop Laptop Quit

Please enter the initial of your desired operation: l

Please enter laptop manufacturer: Dell

Please enter laptop model: New XPS 15 7590

Please enter laptop weight in pounds: 4

Apple MacBook Pro (4.0 lbs.)

vs.

Dell New XPS 15 7590 (4.0 lbs.)

Press any key to continue . . .