# Assignment Description

For this program, you will modify the program you created for [M06 Part A - Programming Assignment 1](https://ivylearn.ivytech.edu/courses/1268733/assignments/19982695?wrap=1). We want to expand the program so that more than one custom car order can be entered at a time. You will modify the main to display a menu of options. The first option will be to read an existing order. The second option will be to add a new order. The third option will be to quit.

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/b9935ed702e6b44a4fd2c534533ba8423336021b/M06%20Part%20B%20Programming%20Assignment%201](https://github.com/wesleyhixon/Programming-Assignments/tree/b9935ed702e6b44a4fd2c534533ba8423336021b/M06 Part B Programming Assignment 1)

# Readme Documentation

Input Information: User can select from 3 menu options to either read an existing order, make a new order, or exit. To read an order, user will be prompted for the order name. To make a new order, user will be prompted for a name and 3 colors for each part of the car.

Output Information: Program outputs a menu with 3 options. If user wants to read an order, the program will find and output the contents of the order’s file, if it exists. If the user wants to create a new order, the program will output a list of colors to choose from for each of 3 parts of the car. Then, the program will save up to 10 orders each to its own order file.

# Flowchart Screen Shots (optional)

Screen shot(s) here

# UML and Use Case Diagrams (optional)

Screen shot(s) here

# Source Code of All files (.h, .cpp)

#include *<iostream>*

#include *<fstream>*

#include *<algorithm>*

**using** **namespace** **std**;

*/\**

*Program Name: Custom Cars Continued*

*Author: Wesley Hixon*

*Date Last Updated: 07/08/2024*

*Purpose: Program which can generate customer orders for car part colors.*

*Functionality added to input up to 10 customers, find a previous order, and output all orders to file*

*\*/*

*// Declaring my enum with every color*

**enum** **carColorType**{SUNSET, CHERRY, PHANTOM, TITANIUM, GEAUX, LIGHTNING, FOREST, MIDNIGHT, PASSION, ROOT, STORM, OCEAN, GLACIAL};

string customerNames[10]; *// String array containing customer last names*

carColorType partColors[10][3]; *// enum array containing 3 part colors for every customer*

int customerIndex = 0; *// Customer index to keep track of which customer we're on*

*// Declaring my function prototypes before main()*

carColorType getColor(string part); *// Returns a color after prompting user*

string printColor(carColorType color); *// Returns a color string given a carColorType color*

void addNewOrder(); *// Add a new order*

int menu(); *// Display menu and return a menu choice*

bool continuePrompt(); *// Asks if user would like to continue*

void quit(int totalCustomers); *// Saves all orders in personalized files and quits*

void readExistingOrder(); *// Prompts user for a name and finds the order*

int main(){

int menuChoice;

bool running = true;

cout << "Welcome to the car customizer!" << endl; *// Welcoming my wonderful user*

**while**(running){

menuChoice = menu(); *// Get menu choice*

**switch**(menuChoice){

**case** 1:

*// Find existing order*

readExistingOrder();

**break**;

**case** 2:

*// Add new order*

**if**(customerIndex == 10){ *// 10 is maximum amount of orders*

cout << "You have entered the maximum number of orders, the program will now save all orders and quit." << endl;

quit(customerIndex);

**break**;

}

addNewOrder();

**break**;

**case** 3:

*// Save and quit*

quit(customerIndex);

running = false;

**break**;

}

**if**(running){

running = continuePrompt();

}

} *// End of loop*

**return** 0;

}

void quit(int totalCustomers){ *// This module saves every customer order before quitting*

cout << "Thank you for using the car customizer. New orders will be saved in files named [customerLastName].txt" << endl;

**if**(totalCustomers == 0){ *// In case of no customers, just quit.*

**return**;

}

**for**(int customerIndex = 0; customerIndex < totalCustomers; customerIndex++){

string fileName = customerNames[customerIndex] + ".txt";

ofstream outputFile;

*// Outputting status*

cout << "Saving " << fileName << "..." << endl;

outputFile.open(fileName); *// Creating file based on customer name*

*// Getting part colors for each part*

carColorType topColor = partColors[customerIndex][0];

carColorType bodyColor = partColors[customerIndex][1];

carColorType trimColor = partColors[customerIndex][2];

*// Saving each color code*

outputFile << topColor << endl;

outputFile << bodyColor << endl;

outputFile << trimColor << endl;

*// Closing the file*

outputFile.close();

}

**return**;

}

void readExistingOrder(){ *// This function prompts for a last name, finds the order file associated with the name, and reads the contents*

string lastName;

cout << "Enter the customer last name for the order" << endl; *// Prompting user*

bool valid = false;

**while**(!valid){ *// Getting valid last name input*

cin >> lastName;

**if**(!cin){

cout << "Please enter a valid input." << endl;

cin.clear();

cin.ignore(10000, '\n');

}

**else**{

valid = true;

}

}

int foundCustomerIndex = 0;

ifstream customerFile;

string fileName = lastName + ".txt";

customerFile.open(fileName);*// Opening file*

**if**(customerFile.fail()){*// In case file wasn't found*

cout << "Customer file for " << lastName << " was not found." << endl;

customerFile.clear();

customerFile.close();

**return**;

}

**else**{ *// If file is found,*

cout << "Here is the order for " << lastName << endl;

string line;

int partType = 0;

**while**(getline(customerFile, line)){ *// Read file line by line*

int partNum = stoi(line); *// Convert the string to an int*

carColorType part = **static\_cast**<carColorType>(partNum); *// Static cast the int to a carColorType*

**switch**(partType){ *// Print the part color*

**case** 0:

cout << "Top color: " << printColor(part) << endl;

**break**;

**case** 1:

cout << "Body color: " << printColor(part) << endl;

**break**;

**case** 2:

cout << "Trim color: " << printColor(part) << endl;

}

partType++; *// Iterate the part type*

}

customerFile.close(); *// Once finished, close the file*

**return**;

}

}

bool continuePrompt(){

char userInput;

cout << "Would you like to continue with your order? Type Y or N: " << endl; *// Asking if the user would like to continue ordering*

bool valid = false;

**while**(!valid){

cin >> userInput;

**if**(userInput == 'Y' || userInput == 'y'){

**return** true; *// If yes, return true*

}

**else** **if**(!cin){ *// In case of input failure*

cout << "Please enter Y or N: " << endl;

cin.clear();

cin.ignore(10000, '\n');

}

**else** **if**(userInput == 'N' || userInput == 'n'){ *// Otherwise, exit and return false*

quit(customerIndex);

**return** false;

}

**else**{

cout << "Please enter Y or N: " << endl;

}

}

**return** 0;

}

int menu(){ *// Menu function, outputs menu and gets input*

int userInput;

cout << "Please choose an option from the menu:" << endl *// Output menu*

<< "1. Read Existing Order" << endl

<< "2. Add New Order" << endl

<< "3. Save & Quit" << endl;

bool valid = false;

**while**(!valid){ *// Validate input*

cin >> userInput;

**if**(userInput >= 1 && userInput <= 3){ *// If input valid, break loop and return input*

valid = true;

}

**else** **if**(!cin){

cout << "Please enter a number between 1 and 3." << endl;

cin.clear();

cin.ignore(100000, '\n');

}

**else**{

cout << "Please enter a number between 1 and 3." << endl;

}

}

**return** userInput;

}

void addNewOrder(){ *// This function adds a new order by getting last name, part colors, and then incrementing customerIndex*

carColorType bodyColor, topColor, trimColor;

string lastName;

bodyColor = partColors[customerIndex][0];

topColor = partColors[customerIndex][1];

trimColor = partColors[customerIndex][2];

cout << "Enter the customer last name for the order: ";

bool valid = false; *// Getting valid user input*

**while**(!valid){

cin >> lastName;

**if**(!cin){

cout << endl << "Please enter a valid last name.";

cin.clear();

cin.ignore(10000, '\n');

}

**else** {

customerNames[customerIndex] = lastName;

valid = true;

}

}

bodyColor = getColor("body"); *// Getting orders for each car part*

topColor = getColor("top");

trimColor = getColor("trim");

partColors[customerIndex][0] = bodyColor;

partColors[customerIndex][1] = topColor;

partColors[customerIndex][2] = trimColor;

cout << "You have chosen " << printColor(bodyColor) << " for the body, " *// Outputting what the user input*

<< printColor(topColor) << " for the top, and " << printColor(trimColor) << " for the trim." << endl;

*// Increment customerIndex to move on to next customer*

customerIndex += 1;

}

carColorType getColor(string part){

*// This prompts user for the color of a part indicated by parameter*

*// It will then use a switch statement to return a carColorType*

int userInput;

cout << "Please choose a color for the " << part << endl; *// Outputting Menu*

cout << "1. Sunset Orange" << endl;

cout << "2. Cherry Bomb" << endl;

cout << "3. Phantom Black" << endl;

cout << "4. Titanium Silver" << endl;

cout << "5. Geaux Gold" << endl;

cout << "6. Lightning Yellow" << endl;

cout << "7. Forest Green" << endl;

cout << "8. Midnight Blue" << endl;

cout << "9. Passion Purple" << endl;

cout << "10. Root Beer" << endl;

cout << "11. Storm Surge" << endl;

cout << "12. Ocean's Rip" << endl;

cout << "13. Glacial White" << endl;

bool valid = false; *// Verifying user input*

**while**(!valid){

cin >> userInput;

**if**(userInput > 0 && userInput <= 13){

valid = true;

}

**else** **if**(!cin){

cin.clear();

cin.ignore(100000, '\n');

cout << "Try again. Please enter a number between 1 and 13." << endl;

}

**else**{

cout << "Try again. Please enter a number between 1 and 13." << endl;

}

}

**switch**(userInput){ *// Switch statement converting int to carColorType*

**case** 1:

**return** SUNSET;

**break**;

**case** 2:

**return** CHERRY;

**break**;

**case** 3:

**return** PHANTOM;

**break**;

**case** 4:

**return** TITANIUM;

**break**;

**case** 5:

**return** GEAUX;

**break**;

**case** 6:

**return** LIGHTNING;

**break**;

**case** 7:

**return** FOREST;

**break**;

**case** 8:

**return** MIDNIGHT;

**break**;

**case** 9:

**return** PASSION;

**break**;

**case** 10:

**return** ROOT;

**break**;

**case** 11:

**return** STORM;

**break**;

**case** 12:

**return** OCEAN;

**break**;

**case** 13:

**return** GLACIAL;

**break**;

}

}

string printColor(carColorType color){

*// Switch statement converting carColorType to string*

**switch**(color){

**case** SUNSET:

**return** "Sunset Orange";

**break**;

**case** CHERRY:

**return** "Cherry Bomb";

**break**;

**case** PHANTOM:

**return** "Phantom Black";

**break**;

**case** TITANIUM:

**return** "Titanium Silver";

**break**;

**case** GEAUX:

**return** "Geaux Gold";

**break**;

**case** LIGHTNING:

**return** "Lightning Yellow";

**break**;

**case** FOREST:

**return** "Forest Green";

**break**;

**case** MIDNIGHT:

**return** "Midnight Blue";

**break**;

**case** PASSION:

**return** "Passion Purple";

**break**;

**case** ROOT:

**return** "Root Beer";

**break**;

**case** STORM:

**return** "Storm Surge";

**break**;

**case** OCEAN:

**return** "Ocean's Rip";

**break**;

**case** GLACIAL:

**return** "Glacial White";

**break**;

}

}

# Three Use Case Screen Shots











