# Assignment Description

You are creating a program that processes custom car orders for a car dealership. You will create an enum that will represent colors used in the car.  For a custom car order, the customer can pick 3 colors, the top, main body, and trim color.

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/d99dece0bd9ba58d56d9639d9e78121d444a4841/M06%20Part%20A%20Programming%20Assignment%201](https://github.com/wesleyhixon/Programming-Assignments/tree/d99dece0bd9ba58d56d9639d9e78121d444a4841/M06 Part A Programming Assignment 1)

# Readme Documentation

Input Information: User will input choices for colors for 3 different body types. They will then decide if they would like those output to a .txt file

Output Information: Output choices for colors for each body type. Then, if the user decides to, output these choices to a .txt 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>*

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

*/\**

*Program Name: Custom Cars*

*Author: Wesley Hixon*

*Date Last Updated: 07/08/2024*

*Purpose: Get a custom car order using enum datatype*

*\*/*

*// Declaring my enum with every color*

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

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

void outputToFile(carColorType bodyColor, carColorType topColor, carColorType trimColor);

carColorType getColor(string part);

string printColor(carColorType color);

int main(){

*// Prompt for colors for body, top and trim.*

*// Then output the colors the user has chosen.*

*// Ask if they'd like to continue with their order*

*// If yes, output the colors to a .txt file*

carColorType bodyColor, topColor, trimColor;

char userInput;

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

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

topColor = getColor("top");

trimColor = getColor("trim");

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;

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

bool valid = false;

**while**(**not** valid){ *// If yes, output colors to a file called order.txt*

cin >> userInput;

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

outputToFile(bodyColor, topColor, trimColor);

valid = true;

}

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

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

cin.clear();

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

}

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

valid = true;

}

**else**{

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

}

}

**return** 0;

}

void outputToFile(carColorType bodyColor, carColorType topColor, carColorType trimColor){

*// Input is the body, top, and trim color enums, output is a file with each line containing the number corresponding to the color*

ofstream outputFile;

outputFile.open("order.txt");

outputFile << bodyColor << endl;

outputFile << topColor << endl;

outputFile << trimColor << endl;

outputFile.close();

**return**;

}

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**(**not** 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





