//

// main.cpp

// as9

//

// Created by Jeff on 10/10/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

// COMSC-200

// Assignment 9 Car Class starter

// Modified from Chapter 13, Programming Challenge 3: Car Class

// A simple Car Class exercise file

// IDE: Codeblocks, Pocketcpp, (g++ compatible)

//

// Test Strings: +,+,+,+,+,+,-,-,.,.,+,.,\_,\_,\_,\_,|

// Test Result: 300 ft

// Test Drive Script for 123 ft:

// Test Drive Script for 321 ft:

// Test Drive Script for 888 ft:

//

#include <iostream>

#include <vector>

#include <string>

using namespace std;

// Car class declaration

class Car

{

private:

string year; // The car's year

string make; // The car's make

int speed; // The car's current speed

int distance; // The car's odometer

public:

/\* Constructor \*/

Car(string y, string m) {

year = y;

make = m;

speed = 0;

distance = 0;

}

/\* Mutator functions \*/

void resetDistance() { // reset (distance) odometer

distance = 0;

}

//--------- to be filled section 1, driving methods

void increaseSpeed() { // accelearate speed function

speed += 5;

if(speed > 30) speed = 30;

distance += speed;

}

void reduceSpeed() { // decelearate speed function

speed -=5;

if(speed<5) speed = 5;

distance +=speed;

}

void cruiseSpeed() { // cruise speed function

speed += 0;

distance += speed;

}

void looseSpeed() { // loose speed function

speed -=2;

if(speed<2) speed = 2;

distance +=speed;

}

void brakeSpeed() { // brake function

speed = 0;

distance += speed;

}

//------end of to be filled section 1------

// Accessor functions

string getYear() { // get year

return year;

}

string getMake() { // get make

return make;

}

int getSpeed() { // get speed

return speed;

}

int getDistance() { // read (distance) odometer

return distance;

}

// public methods

void showStatus() {

cout << " " << distance << " ft @ " << speed << " fps\n";

}

}; // end of class declaration

// function prototypes

void parse(string input, vector<string> &tokens, string del);

int tokenSize(string input, string del);

void strLower(string &str);

void menu(Car c) {

cout << "\n=== Lab10 - " << c.getYear() << " " << c.getMake() << " Test Drive Menu ===\n";

c.showStatus();

cout << " + to accelerate\n"

<< " - to decelerate\n"

<< " \_ to hold speed\n"

<< " . to loose speed\n"

<< " | to break\n"

<< " auto drive instructions\n"

<< " show distance drove\n"

<< " reset odometer\n"

<< " help - this menu\n"

<< " quit - this program\n";

}

int main()

{

cout << "\nAssignment 12"

<< "\nStudent Name:boli zhang "

<< "\nStudent ID:1300601 "

<< "\nFile: car.cpp "

<< "\nTest Script 123 ft: "

<< "\nTest Script 321 ft: "

<< "\nTest Script 888 ft: "

<< "\n\n";

string make;

string year;

//Enter the test drive car info:

cout << "\_\_\_\_setup the test car information\_\_\_\_\n"

<< " Make of this test car? ";

getline(cin, make);

cout << " Year of this test car? ";

getline(cin, year);

Car test\_car(year, make);// Create a test car object.

menu(test\_car);

// main loop

while(true){

string option; // user command

cout << "\n--- Enter a command -> ";

getline(cin, option);

strLower(option);

if(option == "help") {

menu(test\_car);

}

else if(option == "show") {

cout << "\n Odometer:";

test\_car.showStatus();

}

else if(option == "reset") {

test\_car.resetDistance();

test\_car.showStatus();

}

else if(option == "+") {

test\_car.increaseSpeed();

test\_car.showStatus();

}

else if(option == "-") {

test\_car.reduceSpeed();

test\_car.showStatus();

}

else if(option == "\_") {

test\_car.cruiseSpeed();

test\_car.showStatus();

}

else if(option == ".") {

test\_car.looseSpeed();

test\_car.showStatus();

}

else if(option == "|") {

test\_car.brakeSpeed();

test\_car.showStatus();

}

else if(option == "auto") {

cout << "\nEnter the (,) delimited auto drive (+-\_.|) commands: ";

vector<string> tokens;

string input, del=",";

getline(cin, input);

parse(input, tokens, del);

int tSize = tokenSize(input,del);

// cout << "token : " << tokens[1] << "input: " << input[2];

//--------- to be filled section 2, auto pilot script

for(int i =0; i<tSize; i++) {

option = tokens[i];

if(option == "+") {

test\_car.increaseSpeed();

test\_car.showStatus();

}

else if(option == "-") {

test\_car.reduceSpeed();

test\_car.showStatus();

}

else if(option == "\_") {

test\_car.cruiseSpeed();

test\_car.showStatus();

}

else if(option == ".") {

test\_car.looseSpeed();

test\_car.showStatus();

}

else if(option == "|") {

test\_car.brakeSpeed();

test\_car.showStatus();

}

}

//----------end of to be filled section 2

}

else if(option == "quit") {

break;

}

else {

cout << endl<< "INVALID INPUT " ;

}

} // end main menu while

return 0;

}

void parse(string input, vector<string> &tokens, string del) {

tokens.clear();

int countDel =0;

for(int i = 0; i< input.length(); i++){

if(input[i] == del[0])

countDel++;

}

int tokenSize = countDel +1;

for(int i=0; i< tokenSize; i++){

int x= input.find(del[0]);

tokens.push\_back(input.substr(0,x));

input = input.substr(x+1);

}

}

int tokenSize(string input, string del) {

int size;

int countDel =0;

for(int i = 0; i< input.length(); i++){

if(input[i] == del[0])

countDel++;

}

return size = countDel + 1;

}

// String lower case conversion helper

void strLower(string &str) {

locale loc;

string result;

for (int i=0; i<str.length(); ++i) {

result.push\_back(tolower(str[i],loc));

}

str = result;

}

../../../Desktop/Screen%20Shot%202016-10-17%20at%2011.33.11%2../../../Desktop/Screen%20Shot%202016-10-17%20at%2011.37.17%2../../../Desktop/Screen%20Shot%202016-10-17%20at%2011.38.23%2

