Out first class – the Car class

1. Open a new Project in Visual Studio. Name it CarClass.
2. Right click on the Header files and add a new Item, it will be a .h file and call it CarClass.h
3. We will write the Car class declaration in this file.
   1. First, put into some header comments with your name, email and the file name.
   2. Since it is a .h file, we need to put in include guards

#ifndef CAR\_H

#define CAR\_H

#endif

1. Now let’s write the class declaration:

class Car

{

private: // Now we add our access modifier:

string make{ "Mercedes" };

string model{ "S450" };

int year{ 2020 };

string color{ "Black" };

public: //Now add the public access modifier for our methods and constructor

Car(); //default constructor

Car(string make, string model, int year, string color); //overloaded constructor

~Car(); //destructor

void SetCarInfo(string make, string model, int year, string color);

string GetCarInfo*();*

*};*

1. *That’s it. We have declared* our class. Now we have to write the implementation of each method.
2. Open the file, named Car.cpp.
   1. put into some header comments with your name, email and the file name.
   2. Also notice that Visual Studio added #include “Car.h” We need the declaration and the definition together to make a class.
   3. Implement the methods, they are all defined by Car:: to give them scope in the class.

Car:: Circle() - the default constructor. We initialized our variables in the declaration, so there is nothing more to do here.

//Overloaded constructor

Car::Car(string make, string model, int year, string color):

make(make), model(model), year(year), color(color)

{}

Car::~Car() //the destructor, no action in this class

{}

//Set method

void Car::SetCarInfo(string make, string model, int year, string color)

{

this->make = make;

this->model = model;

this->year = year;

this->color = color;

}

//Accessor to get a formatted string

string Car::GetCarInfo()

{

stringstream ss;

ss << "\n The car: " << "\n Make: " << make

<< "\n Model: " << model

<< "\n Year: " << year

<< "\n Color: " << color;

return ss.str();

}

1. And we have completed our class! But it doesn’t do anything by itself. We need to create a test file so we can create and object and use it.
   1. Right-click on the Source Files and add a.cpp file, called Driver.cpp.
      1. Add a header in comments at the top.
      2. We will be inferfacing with the user, so we will #include <iostream> and using namespace std; for convenience.
      3. int main()
         1. cout to the user the nature of the project and what it will do.
         2. create some variables

string make{ "Mercedes" };

string model{ "S450" };

int year{ 2020 };

string color{ "Silver" };

* + - 1. Create a Car object using the default constructor, then display its Info

Car car0;

cout << car0.GetCarInfo();

* + - 1. Create another car object, using the default constructor again

Car car1;

* + - 1. Ask the user for the make, model, year and color of their car.

cout << "\n Please enter the make of your car: ";

getline(cin, make);

cout << "\n Please enter the model of your car: ";

getline(cin, model);

cout << "\n Please enter the year of your car: ";

cin>> year;

cin.ignore();

cout << "\n Please enter the color of your car: ";

getline(cin, color);

* + - 1. Set the user values into the object
         1. car1.SetCarInfo(make, model, year, color);
      2. Get the car info and display to the user
         1. cout << car1.GetCarInfo();
      3. Create a Car object using the overloaded constructor and the same data
         1. Car car2{ make, model, year, color };
      4. Display the car info
         1. cout << car2.GetCarInfo();
      5. Write a good-bye message
         1. cout << "\n Don't you like making cars with use in C++?";
      6. return 0; //end the program

1. Run your program.