**CIS 246 – Spring 2020**

|  |  |
| --- | --- |
| **Class Project:** | **2** |
| **Points:** | **10** |
| **Chapter(s):** | **3** |
| **File(s) to Submit:** | **main.cpp, Car.h (as one zip)** |
| **Due:** | **February 5, 11:00 am** |

**Description**

Write a C++ program that includes a driver program and a class, and prompts the user for some data, saves the data, and prints the data.

**Instructions**

**Open Dev-Cpp and verify these settings (if necessary):**

1. Tools > Compiler Options
2. “Compiler set to configure box” 🡪 **TDM-GCC 4.9.2 64-bit Release**
3. “Add the following commands when calling the compiler” checked
4. Middle box 🡪 **-std=c++11**
5. “Add the following commands when calling the linker” checked
6. Bottom box: 🡪 **-static-libgcc**

**Create a New C++ Console Application Project**

1. Use the file that was been created automatically (**main.cpp**) as the driver
2. An include statement has been added automatically for the **iostream** library. Below this, add the include statement for **Car.h**. Remember to use double-quotes instead of the angle brackets.
3. Add a using directive for the **std namespace**.
4. Save main.cpp for now.

**Add another file to this project:**

1. Right-click the name of the project in the Project browser pane (which is on the left side of the Dev-Cpp window) > click New File
2. Save this new file by clicking the Save icon > name it **Car**. Be sure to change the “Save as type” option to **header files** (.h)!

**Add the code to define the Car class to this file:**

1. Type: **class Car {**
2. The closing curly brace and the terminating semicolon will be added automatically.
3. The rest of the code for this class will go inside these curly braces.

**Add a private data member:**

1. Type **private:** on a line by itself.
2. Below this line, create a string variable called name. Write the data type like this: **std::string** (because we are not using a namespace directive in this file.)

**public get and set member functions:**

1. ABOVE the word private, but still within the class, type **public:** on a line by itself.
2. Below this line, create a **setName** function that uses one string parameter to set the value of the **name** data member;
3. Below the set, create a **getName** function that will return the value of the **name** data member.

**Save Car.h**

**In main.cpp, write the code in the main function:**

1. Create a Car object called **car**.
2. Create a string variable called **carName**. Remember that std:: is not needed here, due to the using directive.
3. Write the statement to prompt the user to type in the name of a car.
4. Write the statement to read what the user types and save it in the variable **carName**.
5. Call the **setName** function of the Car object, and pass the variable **carName**.
6. Display the name of the car by calling the Car object’s getName function.

**Compile and test:**

1. To compile in Dev-Cpp, use the Execute menu, or press the F9 key, or click the icon in the toolbar that looks like four squares in color.
2. After compilation is successful, run the program via the Execute menu, or by pressing the F10 key, or by clicking the icon that looks like a white square to the right of the Compile icon.
3. You can Compile and Run in one step via the Execute menu, or by pressing the F11 key, or by clicking the icon that looks like a square with four colors in it, to the right of the Run icon.

Once the program is working, you can submit it.

**To Submit:**

1. Close Dev-Cpp and save anything if prompted.
2. Locate the main.cpp and Car.h files on disk. Send both to a compressed folder. Use the default name for the zip or name it whatever you like.
3. **Submit only the zipped file** to the dropbox on Canvas. Do not submit any other file(s) except main.cpp and Car.h, within a zip. The zip should not include a folder structure, or DevCpp project files.