

INTRODUCTION TO C++ PROGRAMMING

LABORATORY GUIDE

LAB 1 CLASSES & OBJECTS

By Yidong Cui (cyd@bupt.edu.cn)

School of Software Engineering
Beijing University of Posts and Telecommunications
2019

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



1. Introduction

Based on EGE graphics library which were learned in "C Programming Language", practicing "classes & objects" in C++ with C++11 standards.

This lab will take you 2 hours in class and about 2 hours after classes.

2. Aim of this lab

You should master the usage of a certain C++ IDE and you should finish this lab independently. You will comprehend the concepts of "classes" and "objects" in C++.

You should define and implement the assigned classes which encapsulate the functions in EGE graphics library. You also should use these classes to modify or re-implement your "drawing board" application which you have done in "C Programming Language" course in the previous semester.

3. Contents of this lab

3.1. Create no less than 6 C++ classes

These classes will encapsulate the necessary functions in EGE graphics library.

- 1) A controller class for the graphical environment. This class will be responsible for initializing the graphical environment, destroying the environment, setting the dimensions of the graphical window and etc.
- 2) A point class. This class represents the 2d coordinates of a point.
- 3) A color class. This class represents the colors that can be used in the graphical environment. A color class includes 3 components: R for red, G for green and B for blue. You may add a member of "color_t" type to the class. However, if you do so, you need to add converter functions for converting "RGB" and "color_t".
- 4) A circle class. There should be at least 3 members in this class: the coordinates of the center, the radius and the colors.
- 5) A rectangle class. There must be at least two categories of members in this class: the coordinates which decide the location and size of the rectangle, the colors.
- 6) A triangle class. You can determine the details of the class.
- 7) Other classes as needed. For example, a line class, a polygon class, an arc class and etc.

3.2. Other requirements for the classes

- 1) You should use the point class to represent the coordinates if possible.
- 2) You should use color class to represent the colors.
- 3) There should be at least two color members in any closed geometric shape class. One represents the border color and another represents the fill color.

- 4) There should be one bool member in any closed geometric shape class to indicate whether the shape will be filled with a certain color.
- 5) There should be setter and getter functions for the member data in the class.
- 6) Provide at least two constructors for each class: one default/no-arg constructor and one constructor with arguments.

3.3. General requirements of your program

- 1) Users may input data of geometric shapes, and then these shapes are drawn in the graphics window.
- 2) Users may clear the drawing area
- 3) You may add and implement more functions in your program, like animations, mouse operations, etc.
- 4) You may determine any details that are not mentioned above, provided that these details are reasonable for this lab.

4. Considerations for software engineering

- 1) You need to put all your codes into one "project", no matter which IDE you use.
- 2) Each class should be declared in one header file and implemented in the corresponding cpp file.
- 3) You should name your variables, functions, classes according to the coding rules we have learned in this class.
- 4) You should provide comments for each class. You also should provide comments for some important functions or code block as needed.

5. File version control

You **MUST** use version control system to keep your source codes.

Git is recommended. However, svn or cvs also work, even there are fewer and fewer developers use them.

You may also manage your code with either github.com or gitlab.com. Anyway, please feel free if you prefer other online code repositories, such as gitee.com or www.huaweicloud.com/devcloud/

5.1. What you need to do

- 1) Create a repository, either on local disks or on remote online servers
- 2) Add all your source codes into the repository, including the project configuration files

- 3) **BUT**, do **NOT** add those intermediate or temporary files into the repository.
- 4) Do **NOT** add the exe files into the repository
- 5) Each time, when you finished a function or a class, check in your source codes.

5.2. Recommended tools

You may install “git for windows” (<https://gitforwindows.org/>) first and then TortoiseGit (<https://tortoisegit.org/download/>). TortoiseGit has a Chinese translation, see “Language Packs” section in tortoisegit.org/download

6. The tools and environment

You should develop your program on Windows, because the EGE graphics library are not usable on other OS platforms. **If you insist working on Linux or MacOS, you need to contact the teacher to negotiate a feasible solution.**

You may use IDEs like Visual Studio, Eclipse CDT or Code::Blocks. **The most up-to-date versions** are recommended.

If you use Visual Studio, you should install the community release. As for current situation, you do **NOT** need professional or enterprise releases. Particularly, we **OPPOSE** the using of pirate software when studying how to develop software.

Now, for the EGE graphics library, **only Visual Studio is officially supported**. If you are going to use other IDEs, you need to solve the related issues by yourself.

You are NOT allowed use dev-cpp or visual c++ 6.0, because they are old-enough and will not work with C++11/C++17

7. Submission guidelines

You need to submit the following materials

- 1) The project with your source files. You must do a complete "**CLEAN**" before you pack your project files.
- 2) The executable file.
- 3) The lab report. (Must written according to the report template)
- 4) You should add a "readme" file in your project to demonstrate the information of the programmer, the running environment and the description of the result.

All the above files should be packed into a "zip" file. "rar" compress format is NOT allowed.

When packing your project files, "7zip", an open source compression software, is recommended.

The name of the zip file should be lab prefix plus your school-number plus your name. For example: Lab1-2017211888-张三.zip