

UAH Electrical and Computer Engineering Department
Project02 (3 points) Version 1.0

Submit Your Solution to the Canvas Dropbox by 5:00pm on Friday, February 9, 2018

Important Note

All CPE 212 projects are *automatically graded* in order to provide you timely feedback. So, it is critical that you follow all directions for the preparation and submission of your projects for grading.

Failure to follow the directions may result in zero credit (0 points).

Hint

Match the output of your program to that of the sample solution!!!

Directions for running the sample solution appear on the following pages.

Directions for running the preview script appear on the following pages.

Project02 Goals

A fundamental software engineering skill is the design, implementation, and testing of a software component that must be integrated into a larger software product. In order to do this, the software component must conform to a previously agreed upon interface format. As part of this assignment, you will practice this skill by completing the specification and implementation of **several classes** that will be integrated into a larger software product provided by the instructor. Along the way you will review basic C++ programming and Linux file management skills required for successful completion of CPE 212.

Project02 Overview

For this project, you will *complete* the provided partial C++ program that utilizes classes and inheritance to model geometric shapes. Class **Shape** is an abstract base class used to model properties common to all shapes including its **Color**. **Shape** includes a function (**area**) that must exhibit polymorphic behavior in any derived classes. Class **Triangle** is a derived class that is used to model properties of all triangles. The **Triangle** class inherits from the base class **Shape**. Class **Rectangle** is a derived class that is used to model properties of all rectangles. The **Rectangle** class inherits from the base class **Shape**. The **Project02_Materials.zip** file includes four source code files and a **makefile**.

| Filename | Project02_Materials.zip | Description |
|----------------------|-----------------------------------|--------------------------------------------------------------------------------------|
| main.cpp | Included | Driver program to test Shape , Triangle , and Rectangle classes |
| shape.h | Write and submit this file | Specification file for Shape class |
| shape.cpp | Included | Implementation file for Shape class |
| triangle.h | Included | Specification file for Triangle class |
| triangle.cpp | Write and submit this file | Implementation file for Triangle class |
| rectangle.h | Included | Specification file for Rectangle class |
| rectangle.cpp | Write and submit this file | Implementation file for Rectangle class |
| color.h | Included | Specification file for Color class |
| color.cpp | Write and submit this file | Implementation file for Color class |
| p02input1.txt | Included | Tests class Shape member functions |
| p02input2.txt | Included | Tests class Triangle member functions |
| p02input3.txt | Included | Tests class Rectangle member functions |

Step #1 - Unzipping Project Materials on blackhawk

Use the Firefox browser to access Canvas and download the **Project02_Materials.zip** file into your **Project02** directory. At terminal window prompt, use the unzip utility to uncompress the files. For example, to unzip the files into your current directory:

```
unzip Project02_Materials.zip
```

Since this project is worth three points, you have been given three input files to test your program.

Step #2 – Create Function Stubs for the Missing Support Functions

Open a Linux terminal window and navigate to the directory containing the **Project02** materials downloaded from Canvas (you should be there already if you just completed Step #1 above).

Use the following command to create the missing files by typing the following at the Linux prompt. Linux is case sensitive so pay attention to upper versus lower case letters.

```
gedit color.cpp shape.h triangle.cpp rectangle.cpp &
```

This command starts the **gedit** text editor running as a background process (**&**). With the editor running in the background, you may use the same terminal window to compile and test your program.

-- Place the missing **Shape** class declaration in **shape.h**

-- Place stubs for **Color**, **Triangle** and **Rectangle** functions in **color.cpp**, **triangle.cpp** and **rectangle.cpp** and save the files.

In the terminal window, compile your program by typing the word **make** at the prompt and press the Enter key. If you see error messages, use the text editor to make corrections to the function stubs and switch back to the terminal window to type **make** again to recompile your program. Once your program compiles, you may execute it by typing: **./project02 inputfilename**

Now that your program compiles and executes using function stubs, you may implement and test your solution function by function. Start with the functions associated with opening files and loading data.

Running the Sample Solution on blackhawk [shows the desired output for specified input file]

The best description of what your code should do is the Sample Solution for the project.

Run the sample solution by typing the following at **blackhawk** terminal window command prompt where **inputfilename** is the name of one of the provided input files (for example, p02input1.txt).

```
/home/work/cpe212/project02/p02 inputfilename
```

Your current working directory must contain the input files for this to work.

Running the Preview Script on blackhawk [analyzes outputs for all provided input files]

Run the preview script by typing the following in a **blackhawk** terminal window command prompt

```
/home/work/cpe212data/project02/preview02.bash
```

This script will run both the **Sample Solution** AND your **project02** executable program on the complete set of input files, and it compares the outputs of the two programs line by line to identify errors in your program's outputs. Make sure that the output of your program exactly matches the output of the Sample Solution.

- *To use the preview script, your executable must be on blackhawk*
- *The sample input files must be in your current working directory BEFORE you execute the preview script!!!*