EE3093 Tutorial: C++ week 1

Instructions to users

The exercise in this document can be completed up to 4 degrees of complexity (one for each section); make sure you have understood and completed a section (i.e. you have implemented the code and tested it successfully) section before moving on to the next one.

Part 1: Basic

In **tester.cpp,** function **main** allocates an array of **right_triangle** objects comprising **arraysize** items.

In **tester.cpp,** implement the function: void **Part_1**(right_triangle obj_array[], int arraysize)

The function scans the array and prompts the user to input data for each triangle in the array; then echo this information to screen.

In your implementation, make use of any relevant member function of the class **right_triangle,** found in header file "**RightTriangleExample.h**".

Part 2: Lower Intermediate

In **tester.cpp,** implement the function: void **Part_2**(right_triangle obj_array[], int arraysize)

The function scans the array to identify the entry with the smallest (non-zero) area; then prints to screen the details of this triangle. If more entries are found with the same area, select the one with the lowest index in the array.

In your implementation, make use of any relevant member function of **right_triangle.**

What happens if elements of the array have been initialized with incorrect values for the sides (negative numbers)?

What is the smallest modification that can be applied to your implementation so that, if more entries are found with the same smallest area, the one with the largest index in the array is selected?

Part 3: Higher Intermediate

In header file "**PolygonWcolor.h**" define the class **right_triangleWcolor**; this is a composite object similar to the "Rectangle with colors" discussed in this week's lectures.

Useful public member functions to implement are:

void **inputFromKeyboard**() that allows the user to input data for the triangle with colors from keyboard.

void **printInfo**() that prints to screen data for the triangle with colors.

void **inputRandomValues**(double max_val=100) that assigns random values to the triangle with colors

Use member functions of the constituent objects to implement the functions above; follow similar example discussed in this week's lectures for "Rectangle with colors".

Part 4: Advanced

In **tester.cpp,** implement the function: void **Part_4**(double min_area, polygonColorOptions pcolr_target)

The function allocates **array_1**, an array that holds max_elements of **right_triangleWcolor** objects (set max_elements = 100 to begin with).

The function the scans the array to assign dimensions and color to each element in the array, using the **inputRandomValues** member function.

The function allocates **array_2** of the same type and size of array_1; the function fills array_2 with all the elements of array_1 that have area above a given threshold (input to **Part_4**) and that match a given color (input to **Part_4**). Print the content of **array_2** to screen.