

Assignment 3

Weight: 25% of your final grade

Due: After Unit 13

Assignment 3 requires you to write **three** C++ programs using the information you have studied in Units 8–13.

C++ Programs

1. Write a program in which you create a **const** whose value is determined at runtime by reading the time when the program starts (hint: use the `<ctime>` standard header). In a separate function, have the program create two arrays of 10,000 **doubles**. Initialize the first array with sequential integral values starting with 100, and initialize the second array with the same numbers, but in reverse order (i.e., the first array would contain 100, 101, 102... while the second array contains 10,099, 10,098, 10,097...). Loop through both arrays using a single loop, and multiply the corresponding array elements from each array together and display the result. Read the time when the program completes the multiplication, and compute and display the elapsed time. Do not use inline functions in this program.
2. Rewrite program 1 using an inline function to perform the calculation. In the test plan for this program (actual results section), compare the time required by this program to execute against the time required by the first (non-inline) program.
3. Write a program that creates a class hierarchy for simple geometry.
 - a. Start with a **Point** class to hold x and y values of a point. Overload the `>>` **operator** to print point values and the `+` and `-` **operators** to add and subtract point coordinates (hint: keep x and y separate in the calculation).
 - b. Create a base class **Shape** which will form the basis of your shapes. The **Shape** class will contain functions to calculate area and circumference of the shape, plus provide the coordinates (Points) of a rectangle that encloses the shape (a *bounding box*). These will be overloaded by the derived classes as necessary. Create a **display()** function that will display the name of the class plus all stored information about the class (including area, circumference, and bounding box).

- c. Build the hierarchy by creating the **Shape** classes **Circle**, **Square**, and **Triangle**. For these derived classes, create default constructors and constructors whose arguments can initialize the shapes appropriately using the correct number of **Point** objects (i.e., **Circle** requires a **Point** center and a radius; **Square** requires four **Point** vertices, while **Triangle** requires three **Point** vertices).
- d. In `main()`, create one instance each of the following: a Circle with a radius of 23, a Square with sides 25, and a Triangle with sides 10, 20, 30. Define all of them so that the origin (0,0) is somewhere within each object. Display the information from each object.

Deliverables

The deliverables for this assignment are as per the Assignment Requirements and Marking Criteria document.



Pay special attention to the file-naming convention!

All deliverables must be zipped into one file. Upload the file here to send it to your tutor for marking and feedback.

Marking Scheme

You will be graded on the following criteria:

Program compiles without errors	20%
Correct program execution	20%
C++ program correctness and style	20%
Program documentation	20%
Test plan	20%
Total	100%