Assignment 9 – Classes and Class Templates

In this assignment, you will create some simple class definitions, as well as a class template.

Create class definitions for three different kinds of fruit. You can choose whichever ones you would like. I might choose: strawberries, bananas, and pomegranates. Your class declarations should be placed in a file called Fruit.h, and each one should have: a constructor, a destructor, and a print function. None of these should take any parameters. Each fruit should also have a data field, giving the weight of the fruit, in grams, which should be represented as a floating point number.

In the corresponding source code file, Fruit.cpp, create the function definitions for your three kinds of fruit. Mainly ensure that each function prints a message, for debugging purposes, e.g. “New 100 gram banana!” for the banana constructor, “100 gram banana is alive and well” for the banana print function, and “Bye-bye 100 gram banana!” for the banana destructor function. The constructor function should set the weight of the fruit to a random but reasonable value (no 10 lb strawberries, please), and the messages should all be sure to include the weight of the fruit.

Templates are very useful, in that they allow you to create code in which the types are not fully specified. C++ allows you to create function templates, and class templates. Class templates are most commonly used to create containers. For example, you could use a class template to create an array, but without committing to the contents of the array. The user of the array would be at liberty to specify what the array should contain. Templates are much more versatile than this, although container classes are certainly where class templates find their greatest application. This is their ‘classic’ area of application.

Create a *class template* for a fruit plant, i.e. you are going to create a generic fruit plant, which will not be committed in its code to being one particular type of fruit plant or another. Your fruit plant class template should have: a constructor, a print function, and a destructor. Your plant should contain an array of fruit. Inside the plant’s constructor, you should randomly decide how many pieces of fruit the plant will have, and you should use the C++ keyword new (remember to use the array form), rather than malloc to generate the array. Each plant can only have one type of fruit (i.e. so there won’t be any weird Frankenstein fruit plants, growing both pomegranates and bananas). In the plant’s print function, call the print function for each piece of fruit on the plant. In the plant’s destructor, use the array form of delete to free the array.

Create a function main() to test your function template. Instantiate three plants: one for each fruit type. Call the print function for each. Due to the code you already wrote in the fruit plant and fruit classes, your output should show the creation, printing, and destruction of all the fruit on all the plants.