Short Answer Questions

1. Normally, function bodies do not appear in header files, but they do appear in the header file for our template of dynamic arrays. Why?

By writing the body of the function in the header files of the template allow us to replace the type values in the <class T> argument with our desired data types at compile time. For example, the user can now specify which type of DynamicArray will she create i.e. if T is a double array, then DynamicArray<double> or int then DynamicArray<int>.

This gives us a powerful level of abstraction which allow us the flexibility to create objects of different types (or to execute functions of different return types or argument types) form the same header file template.

1. What are the differences between templates and macros?

A template is specified by using the keyword <template <class T1, class T2, …> and a macro is specified by using the keyword <#define NAME definition>.

Templates parameters are meant to be replaced with an appropriate data type by the user. If the user does not supply the types of data that the template will use, we will get a compile error.

On the other hand, a macro does not need the user to specify a data type. If a constant is defined as a macro, for example #define PI 3.14, then in all instances of the code that uses PI, the preprocessor will replace PI with the value 3.14 and then the compiler will decide the type of the variable PI. The same happens with a function, if the developer defines #define F(x) (x+2), then in all instances of the code that has F(<number>), the preprocessor will replace x+2 with <number>+2, and then the type of F(x) will be decided by the compiler.

Templates give us more control with data types, macros are a simple way to replace values.

1. Discuss the differences between formal parameters to a function and the parameters to a template.

The parameters to a template are asking the user to specify a type of data. On the other hand, a function parameter will ask the user for a data value that is of a particular type.

For example, if you have a function template:

template <class T1, class T2>

T1 function(T2 arg1, T2 arg2){<body>}

The parameters T1 and T2 need to be replaces with an appropriate data type (int, double, string, etc.). Menawhile, the formal parameters of the function need to be data values of type T2.