1. Create a header file (with an extension of ‘**.h**’). In this file, declare a group of functions by varying the argument lists and return values from among the following: **void**, **char**, **int**, and **float**. Now create a **.cpp** file that includes your header file and creates definitions for all of these functions. Each definition should simply print out the function name, argument list, and return type so you know it’s been called. Create a second **.cpp** file that includes your header file and defines **int main ( )**, containing calls to all of your functions. Compile and run your program.

2. Create two functions, one that takes a **string\*** and one that takes a **string&**. Each of these functions should modify the outside **string** object in its own unique way. In **main ()**, create and initialize a **string** object, print it, then pass it to each of the two functions, printing the results.

3. Write a simple class called **Simple** with a constructor that prints something to tell you that it’s been called. In **main( )** make an object of your class Add a destructor to Exercise that prints out a message to tell you that it’s been called. a class contains an **int** member. Modify the constructor so that it takes an **int** argument that it stores in the class member. Both the constructor and destructor should print out the **int** value as part of their message, so you can see the objects as they are created and destroyed.

4. Create a class without any constructors, and show that you can create objects with the default constructor. Now create a non-default constructor (one with an argument) for the class, and try compiling again. Explain what happened.

5. Create a **Text** class that contains a **string** object to hold the text of a file. Give it two constructors: a default constructor and a constructor that takes a **string** argument that is the name of the file to open. When the second constructor is used, open the file and read the contents into the **string** member object. Add a member function **contents ( )** to return the **string** so (for example) it can be printed. In **main ( )**, open a file using **Text** and print the contents.

6. Create a **Message** class with a constructor that takes a single **string** with a default value. Create a private member **string**, and in the constructor simply assign the argument **string** to your internal **string**. Create two overloaded member functions called **print( )**: one that takes no arguments and simply prints the message stored in the object, and one that takes a **string** argument, which it prints in addition to the internal message. Does it make sense to use this approach instead of the one used for the constructor?

7. Determine how to generate assembly output with your compiler, and run experiments to deduce the name decoration scheme.

8. Create a class that contains four member functions, with 0, 1, 2, and 3 **int** arguments, respectively. Create a **main ( )** that makes an object of your class and calls each of the member functions. Now modify the class so it has instead a single member function with all the arguments defaulted. Does this change your **main ( )**?

9. Create a function with two arguments and call it from **main ( )**. Now make one of the arguments a “placeholder” (no identifier) and see if your call in **main ( )** changes.