EE208 Laboratory Session 8

Bryan Hennelly

3rd April 2018

1 Lab objectives

In this lab you will start to work will design and implement some more classes in C++ using different h and cpp files combined within a single project. Having completed this lab you should be very comfortable creating classes. You should understand the details of object construction and destruction within such hierarchiesi in more detail. You will design solutions that use static members, friends, and you will also have to use inheritance for the first time. This lab is due by 5pm Monday 9th April.

2 Questions:

For each of the problems given below write a C++ program that provides a solution.

Remember:

- **Comment your code.**
- **Use proper indentation for function and control structures.**

Exercise 1: Make a class called `Resistor` describing a resistor with the following members:
a private data member for the resistance, R
a public member function to set the value of R
a public member function to calculate the current I from the voltage V
a public member function to calculate the voltage V from the current I
a public member function to calculate the power dissipation P from the current I using the formula P = IV, where V is calculated using the previous member function.
add a constructor to set the value of R when it is defined.
now add a default contructor.

Make a program to test this class called `driver.cpp` that creates two resistors called A and B. The first resistor should have its value set using the constructor while the second is declared using the default constructor and has it's value set using the set_value function you defined. You should then print out all related information for the resisitors (you might need to add another member function to help). Your class declaration should be stored in a file called `resistor.h` and the class definition should be stored in a file called `resistor.cpp`. Finally, expand on your previous code as follows: adapt `driver.cpp` to request a user to enter an integer N and then dynamically allocates an array of type `Resistor` of size N. You should then use a `for` loop to assign random values (doubles) between 0 and 1000 to these N resistors.

Exercise 2: Take your solution from the first exercise and add a static member `int` variable called n that allows you to count the number of resistor objects that are active in the main function. Make sure to give n an initial value of 0 and to increment n by 1 in both of your constructors. You will also need to add a destructor that subtracts 1 from n. Test to see if your counter works by prinitng the value of n. Following this, delete your array (from the first exercise) and again print out the value of n. Finally, declare two resistors A and B and give them some values. Then, create a `friend` function called `series` that takes in two resistor objetcs and returns a resistor object. The R value of the returned resistor should be equal to sum of the R values of the two input resistors. Demonstrate that this function works in your main function. How might you replace this friend functon with a method in your resistor class?

Exercise 3: Create a new class called `circuit`. This class should contain a private member, which is an array of (or more accurately a pointer to) N resistors. The ciruit class should have a constructor that has one input, an `int` N, and which dynamically allocates memory for the N resistors and gives them defualt values between 0 and 1000. The class should also have an appropriate default contructor and a destructor. In your main function (in `driver.cpp`) ask the user to input an int N and use this to creata a circuit object. Print the value of n (the static member variable) to confirm that the appropriate number of resisors have been created. The `circuit` class declaration should be stored in a file called `circuit.h` and the class definition should be stored in a file called `circuit.cpp`.

Exercise 4: And now for an example that will use inheritance. Start again with a new project. Create a class for a book that contains information about the title, author, publisher, and year of publication. The class should have a constuctor that allows you to create an object while entering all of these values. It should also have a set_values function that allows you to enter these values. Create a new class that inherits from this class publicly, called library_book. This new class should have a variable of type bool called on_loan, which indicates whether the book is on loan. It should also have two new function called check_in and check_out which change the value of on_loan appropriately. This new class should have a new constructor that creates a default value for on_loan as well as retaining the functionality of the constructor of book. Demonstrate the use of the library_book class in a main function. Your files should be stored in book.h,book.cpp,library_.book.h, library_.book.cpp and driver.cpp where your main function is locaated. For good housekeeping, put all of these files in a single folder.

Exercise 5: Write two derived classes inheriting functionality of a base class called person (should have a constructor member function that asks the user to enter name and age) and with added unique features of student, and employee, and functionality to assign, change, and delete, records of student and employee. What these unique features are is up to you. Actually from now on you will not be told how to design your solutions - you will be responsible for this. Make sure you ave functionality to print out the details of your object and demonstrate some objects of your derived classes in the main function.