

CS23710 Assessed Assignment 2013-2014

“For those in Peril on the Sea”.

Further Information

David Price and Fred Long

November 12, 2013

1 Accessing the Data and Pre-Compiled Code

This document describes how to access the data files we have provided and how to make use of the pre-compiled “great circle distance” function that we have provided.

2 Data Files

As mentioned in the assignment document, we have provided a set of data files which you are required to process.

They are available from the module’s site on blackboard and in addition, they are also available from the directory

`/dcs/dcsproto/cs23710/data/`

which should be mounted when you log into our Linux Mint computers as installed in room C56.

The directory contains (at present just):

- `rescue_assets.txt`
- `ships_1.txt` this holding the first set of ships
- `mayday_1.txt` this holding the mayday calls relating to the first set of ships

These will be joined by a small group of other files, probably making up six scenarios in total.

3 Our Pre-compiled Function

We have provided one pre-compiled function, in the form of a library, that calculates the distance between two locations.

The library containing the functions is

`/dcs/dcsproto/cs23710/libnavigation.a`

The interfaces of the functions are declared in

`/dcs/dcsproto/cs23710/navigation.h`

4 Using our header file and library from NetBeans

First, create a new NetBeans project, with a sensible name, making sure that you set it to be a C/C++ Application and that it will create a C main file (not C++ nor Fortran).

Now, go to the Header Files for the project and click right button and select to add an existing item. Navigate around the filestore (to `/dcs/dcsproto/cs23710/`) until you locate the `navigation.h` header file we have provided and then click select.

Now right button on the name of your project and select Properties (the last item in long list).

4.1 Compiler Options

Select **Build** => **C Compiler** in left hand section.

Now go to the upper part of the right hand area where it says Include Directories and click to the left of the ... A box should appear and in that box type

`//dcs/dcsproto/cs23710/` and then click Apply.

4.2 Linker Options

NOTE: it is crucial that you add the library file for our supplied code *BEFORE* you add the standard maths library. If you do not do it in this order, then when your program links it will fail to find the maths function that we use.

Now select **Build** => **Linker** in left hand section.

Now go to the lower part of the right hand area where it says Libraries and click on the ... at the right.

In the right hand area click the “Add Library File...” button.

(If needed, navigate to `/dcs/dcsproto/cs23710/` and select that and then click OK.)

It should now show you the contents of `/dcs/dcsproto/cs23710/` which should include `libnavigation.a` so then select that and click OK.

In the right hand area click the “Add Standard Library...” button.

Choose the Mathematics library and click OK.

In the main properties window click Apply and then OK.

5 Now design and write your program

Our code and library are now attached to your project. You now need to plan and design the rest of your program and add appropriate files to the project.

6 Development Suggestions

6.1 Data Files

I suggest you start by looking at the data files that have been supplied and make sure that you understand what they contain.

Having done that, try to write some simple functions that open the files and read in the information. You will need to use functions like `fopen()` and `fscanf()` to open files and read data from them.

Of course, you will need to generate prompts for the use of your program so that they can specify the names of the files that are to be processed. The filenames should not be hard-coded into the final version of your program.

6.2 Our great circle distance function

Now look at the header file that defines the interface to our great circle distance function and the parameters that it uses.

The first struct parameter contains two fields that are the latitude and longitude of one location. The second struct parameter contains two fields that are the latitude and longitude of another location.

The double return value is the calculated great circle distance between them.

Make sure you understand what is described.

7 Our Header File - /dcs/dcsproto/cs23710/navigation.h

```
/*
 * File:   navigation.h
 * Author: dap
 *
 * Created on 11 November 2011, 10:21
 */

#ifndef NAVIGATION_H
#define NAVIGATION_H
/*
 * This structure is used to represent a location.
 * The two members are doubles representing the latitude and longitude
 * of the location in degrees.
 */
typedef struct {
    double lat;
    double lng;
} location;

/* The great_circle function takes two struct parameters
 * and returns a double.
 * The first struct parameter represents one location.
 * The second struct parameter represents a second location.
 * The great_circle function returns the number of nautical miles
 * between two locations as a double.
 */

double great_circle(location location_1, location location_2);

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

#endif /* NAVIGATION_H */
```