

# ENEE 140 Lab 3

## Lab instructions

This handout includes instructions for the recitation sessions on Wednesday and Friday. **Follow these instructions** to familiarize yourself with the debugger from your IDE, then **submit the homework** as indicated below. To prepare for the next lecture, complete the **reading assignment** and try to solve the **weekly challenge**.

### 1 Basic lab skills

Make sure that you know the following by now:

1. Log in your GRACE account from PC and Linux
2. Access class public directory on GRACE
3. Access class webpage at <http://ter.ps/enee140>
4. Access the Piazza message board, at <https://piazza.com/umd/spring2016/enee140>
5. Write, build and run programs in your IDE (CLion or Eclipse)
6. Use the **submit** command to electronically submit your assignment

### 2 Familiarize yourself with the CLion debugger

Watch the video tutorial at <https://www.jetbrains.com/clion/documentation/> and read the more detailed documentation at <https://www.jetbrains.com/clion/help/debugging.html>. Then load the **temperature\_conversion** program in CLion and start debugging it:

1. Set a breakpoint on the line with the **printf** statement.
2. click the “Debug” toolbar button.
3. Run the program step-by-step, by clicking the “Step Over” button repeatedly, and observe how the values of the program variables change at each step.
4. Run the program (it should pause at the breakpoint).
5. Click the “Resume” button repeatedly, and observe how the values of the program variables change at each iteration of the loop.

6. Look at the Console pane, where the program output is shown. At each loop iteration, try to predict the next line that will be printed, given the current values of the program variables.

## Homework

**Due:** February 12 at 11:59 pm.

Create two files, called `bugs_found.txt` and `gas_mileage_conversion.c`, by following the instructions below. Submit it using the following command:

```
submit 2016 spring enee 140 AAAA 3 bugs_found.txt
submit 2016 spring enee 140 AAAA 3 gas_mileage_conversion.c
```

Note: you must replace `AAA` with your own section number (0101, 0102, etc.)

### 1 Debugging

Copy the programs `bugs1.c`, `bugs2.c`, `bugs3.c`, `bugs4.c`, `bugs5.c` from the `public/labs/week03` directory to your own account. For each one of them, create an Eclipse project and add the source code to the project. Debug the five examples as follows:

1. Read the code lines that Eclipse indicates as being incorrect.
2. Think about the reason why these lines are incorrect. These programming errors are called “bugs”.
3. Modify the code to correct the bugs.
4. Compile the modified code.
5. Go back to step a if there are errors and warnings at compilation step d
6. Execute the compiled program
7. Go back to step a if the output is not what you expected

For each piece of the code, write down the bug(s) and correct them in `bugs_found.txt`. (Do not copy the entire code.)

### 2 Gas mileage

In the United States, the fuel efficiency of a car is customarily expressed in miles-per-gallon (mpg) of gasoline, while in European countries it is usually expressed in liters-per-100km (l/100km). Using the programming techniques demonstrated in class for converting Celsius degrees to Fahrenheit degrees, write a program called `gas_mileage_conversion.c` that prints an mpg – l/100km conversion table. The table should start at 10 mpg and end at 50 mpg, with an increment step of 5 mpg.

Remember that:

$$\begin{aligned} \text{Kilometers} &= \text{Miles} * 1.60934 \\ \text{Liters} &= \text{Gallons} * 3.78541 \end{aligned}$$

**Hint:** To determine if your program produces the correct results, you can search Google for a particular conversion, for example “25 mpg in l/100km”.

## Reading assignment

K&R Chapters 1.7, 1.8, 7.2, 7.4, B4

## Weekly challenge

Rewrite the Celsius  $\mapsto$  Fahrenheit temperature conversion program to use a function: implement a function, called `celsius2fahrenheit`, that implements the conversion and invoke this function from the while loop that prints the table. You can use the following template (also available in the GLUE class public directory, under `public/challenges/week03/`):

```
/*
 * temperature_conversion_function.c
 *
 * Rewrite the temperature conversion program to use a function.
 * K&R Exercise 1.15
 */

#include <stdio.h>

#define LOWER    -100
#define UPPER    100
#define STEP     10

float celsius2fahrenheit(float celsius);

int
main()
{
    // Add your code here

    return 0;
}
```

The weekly challenge will not be graded. However, if you manage to solve it, you may submit it for extra credit. The deadline for submitting your solution to the weekly challenge is **Monday at 11:59 pm**. To be eligible for extra credit, you must implement correctly **all but two** of the weekly challenges. You can submit your program from a GRACE machine using the following command (replace `AAAA` with your section number):

```
submit 2016 spring enee 140 AAAA 1003 temperature_conversion_function.c
```