

**CSC112**                      **Fall 2010**  
**Fundamentals of Computer Science**  
**Lab 2**

In this lab, we get started with learning the C++ language. Your assignment consists of several programs which are progressively introduce new language features. Create a directory named Lab2. Keep all of your source and compiled programs in the directory Lab2.

0. Our first program is pretty simple, just to get you writing on your own.

Write a C++ program to input a temperature in Fahrenheit and output the equivalent temperature in Celsius. The formula is:

$$C = 5/9(F - 32)$$

Be mindful of the distinction between integer division and floating point division.

1. Modify your program from part 0 to check for valid input. Use the member function `cin.fail()` to determine if the input is acceptable.
2. Modify your program from part 0 to so that the temperature conversion calculation is done by a function called “fahrenheit\_to\_celsius()”. The function header should be:

```
float fahrenheit_to_celsius(float f)
```

3. Write a function named “is\_prime” with the following header

```
bool is_prime(int k)
```

Your function should return true if `k` is a prime number and return false otherwise. We have not discussed arrays yet, so we will **not** be using the sieve of Eratosthenes method for finding primes. To check if a number is prime,

- (a) If an integer is less than two it is not prime.
  - (b) Check if `k` is greater than two; if so, check if it is divisible by two.
  - (c) for every odd integer `j` from three up to the square root of `k`, check if `k` is divisible by `j`, but stop searching if you find such `j`.
  - (d) If a divisor of `k` was found in step 2 or 3, return *false*, otherwise, return *true*.
4. Write a simple main program that reads a number, calls your function “is\_prime” and outputs the conclusion “prime” or “not prime”. Run your program with several different numbers to convince yourself that `is_prime()` is working correctly. Don’t forget to check the so-called “corner cases”, i.e., for this example, try inputting 2, try again, inputting zero. This process is called “unit testing”. The simple code that allows you to do the testing is sometimes called “scaffolding”.
  5. Using your function `is_prime()`, write a program that prints all prime numbers up to 64.

**Turn in:** Change to the directory containing the sub-directory “Lab2” Create a file named “lab2.tar” using the command:

```
tar cf lab2.tar Lab2
```

Upload the file “lab2.tar” to your account on telesto.