**Coding challenge #1**

Allow the user to enter a temperature in degrees Celsius.

Calculate and output the equivalent temperature in degrees Fahrenheit, formatted to 2 decimal places.

*However, a suitable error message should be output if the initial temperature in degrees Celsius is not valid, i.e. not an integer.*

**e.g. sample on-screen dialogue**

**C:\Users\QA\rust>ctofinput**

**Enter a temperature in Deg C:**

**r**

**r is not a valid temperature**

**C:\Users\QA\rust>ctofinput**

**Enter a temperature in Deg C:**

**100**

**212.00 Deg F**

**Extension task:**

Ensure all warnings are suppressed (Hint: you may need to use an attribute)

**Coding challenge #2**

Write a Rust program which allows the user to enter a whole number in denary (base 10) and which outputs its equivalent value in:

* Binary (base 2)
* Octal (base 8)
* Hexadecimal (base 16)

*However, a suitable error message should be output if the initial value input is not valid, i.e. not a denary integer.*

**e.g. sample on-screen dialogue**

**C:\Users\QA\rust>baseconv**

**Enter a denary number:**

**65**

**Binary: 0b1000001**

**Octal: 0o101**

**Hex: 0x41**

**C:\Users\QA\rust>baseconv**

**Enter a denary number:**

**HELLO**

**HELLO is not a valid integer**

**Coding challenge #3**

Create a (very) simple command line application which will accept two integers, e.g.

C:\Users\QA\rust>commandlinecalc.exe 3 4

And which then prints their sum e.g.

7

**Extension task (using guidance in this chapter)**

Display an appropriate error message if insufficient arguments are provided, e.g.

**no arguments**

**C:\Users\Mark\rust>commandlinecalc.exe**

**Aborting! Insufficient Arguments.**

**Usage: commandlinecalc.exe <int> <int>**

**only 1 argument**

**C:\Users\Mark\rust>commandlinecalc.exe 3**

**Aborting! Insufficient Arguments.**

**Usage: commandlinecalc.exe <int> <int>**