# Assignment 1: Chapter 2, “Data and Expressions” (8%)

This programming project should be completed and submitted by the end of Week 4, and is worth 8% of your final grade. Please refer to the “Assignment Instructions” for details on the marking rubric and submission instructions.

1. Write an application that converts miles into kilometres (one mile equals 1.60935 kilometers). Read the miles value from the user as a floating point value and report the equivalent amount in kilometres.

**Hint:** You should treat the conversion value of 1.60935 like the values in the TempConverter.java example.

**Testing:** Submit testing exhibits using the following inputs: 0, 1, 100, 0.5, and 123.45.

1. Write an application that reads a value representing a number of seconds, then displays the equivalent amount of time as a combination of hours, minutes, and seconds. For example, 9999 seconds is equivalent to 2 hours, 46 minutes, and 39 seconds.

**Hint:** This requires using the % modulus operator. The trick with multiple units of measure like this is to start with the largest unit (hours) and work down.

**Testing:** Submit testing exhibits using the following inputs: 0, 60, 3600, and 9999.

1. Write an application that prompts for and reads the numerator and denominator of a fraction as integers, then displays the decimal equivalent of the fraction.

**Hint:** This requires casting from integer to double before dividing.

**Testing:** Submit testing exhibits using the following inputs:  
1, 2 giving 0.5; 3, 8 giving 0.375; 2, 3 giving 0.666...

|  |  |
| --- | --- |
| **Assignment Marking Criteria** | **Weighting** |
| **Correctness of solution:** Algorithm is implemented and produces correct results for the stated problem | /4 |
| **Testing:** Submission of test exhibits to indicate the solution works for a range of cases (e.g. minimum and maximum inputs) and handles unexpected exceptions | /2 |
| **Comments and documentation:** Source code contains comments that explain in plain English what the code is intended to do  **Note:** Javadoc style is **not** required. | /2 |