Introduction:

Since this is Week 2, I am not going to re-explain all the programming features that I have re-used from last week, such as the import statement of basic class initialisation methods. However, after reading through the lecture notes and other materials, such as ([Bergin et al., 2005](#_ENREF_1)), I have learned how to separate each task into separate methods and then call the methods in turn in the main class. Also, ([Watson, 2012, p. 12](#_ENREF_3)) provided me with an insight of how to round numbers and how to use doubles, while ([Deitel & Deitel, 2012, pp. 109-111](#_ENREF_2)) showed me how I could use IF statements to handle null data entries.

import javax.swing.\*;

public class kilometreConversion{

double inputMetres; // I used doubles to represent these numbers, just in case the

// user wanted to use numbers that needed a decimal point.

String metreEntry; // The reason I used a string command was because I knew I

// would need a text string if the user was to successfully enter

// information.

double kiloMetres;

public static void main(String[ ] args){ // This part calls and executes the methods.

kilometreConversion mykilometreConversion = new kilometreConversion();

mykilometreConversion.GetMetres();

mykilometreConversion.ConvertToKilometres();

}

public void GetMetres(){ // This part takes the number of metres that have been entered, and converts them into a double value. (If nothing has been entered, it returns a zero).

String metreEntry = JOptionPane.showInputDialog("Please enter the number of metres:");

if(metreEntry.length() > 0) {

inputMetres = new Double(metreEntry).doubleValue();

}

else

inputMetres = 0;

}

public void ConvertToKilometres(){ // This part takes whatever the user entered and divides it

// by a thousand to get the number of KMs. It then uses the

// round function to round up the double to the nearest KM.

// If nothing was entered by the user, it takes a zero from the

// last method and divides it by 1000 to get zero.

kiloMetres = inputMetres / 1000;

JOptionPane.showMessageDialog(null, "When rounded up to the nearest kilometre, that is " + Math.round(kiloMetres) + "km.");

}

}

**References**

Bergin, J., Stehlik, M., Roberts, J. & Pattis, R. (2005) *Karel J.Robot: A Gentle Introduction To The Art of Object-Oriented Programming in Java* [Online]. Seidenberg School Of Computer Science And Information Systems. Available from: <http://csis.pace.edu/~bergin/KarelJava2ed/ch2/javamain.html> (Accessed: 14th July 2013).

Deitel, P. & Deitel, H.M. (2012) 'Chapter 4: Control Statements Part I'. In: M. Hirsch (ed.), *Java: How To Program.* 9th edn, Boston, Massachusetts: Prentice Hall.

Watson, K. (2012) *Object-Oriented Programming in Java: Seminar 2* [Online]. Laureate Online Education B.V. Available from: https://elearning.uol.ohecampus.com/bbcswebdav/xid-584611\_4 (Accessed: 14th July 2013).