**Programming Principles 02 Tutorial 02**

**Task 1 – A Simple Program**

Based upon the simple program from the lecture, **write a program that performs a conversion**. Your program should prompt for a value, perform the necessary arithmetic to convert the value to the desired result, and then display the result. I’ve listed some common conversions between Metric and Imperial measurements below, but feel free to google for others and implement them.

**Length**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 centimetre  =  0.393 inches | 1 inch  =  2.54 centimetres | |
|  | 1 metre =  3.281 feet | 1 foot  =  0.305 metres | |
|  | 1 kilometre  =  0.621 miles | 1 mile  =  1.609 kilometres | |
|  | **Weight** |  |  |
|  | 1 gram = 0.035 ounces | 1 ounce = 28.35 grams | |
|  | 1 kilogram = 2.204 pounds | 1 pound = 0.454 kilograms | |
|  | **Volume** |  |  |
|  | 1 millilitre = 0.035 fluid ounces | 1 fluid ounce = 28.413 millilitres | |
|  | 1 litre = 0.878 quarts | 1 quart = 1.137 litres | |

Once you’ve typed the code, **save**, and then **run** the code. The prompt for a value will appear in the console window.

Run the program a few times to **test** it and make sure that it is working as intended. Try entering something that isn’t a number into the prompt and note what occurs (and why).

Write and test separate programs to do different conversions for conversion given in the above table.

**Task 2 – A Slightly Less Simple Program**

For a bit more of a challenge, let’s write a program with slightly more complex processing. The code for this program will still follow the same basic structure of obtaining input, performing arithmetic on it, and then displaying the result, but the conversions/formulas are slightly more advanced. Again, you are welcome to implement different conversions and formulas than the ones below.

**Temperature**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Celsius = (Fahrenheit - 32)\* 5 / 9 | Fahrenheit = Celsius \* 9 / 5 + 32 | |
|  | **Geometry** |  |  |
|  | Area of circle = π \* radius2 | Circumference of circle = 2 \* π \* radius | |

*You can either use 3.1416 as the value of π (pi), or use Math.PI to refer to the value of pi in calculations. The Math class gives your program access to extra mathematical functions and values.*

*Why make two programs when you can make one? After prompting for the circle’s radius, your program can calculate both the area and circumference of the circle and show both results.*

**Nutrition**

Body Mass Index (BMI) = Weight in Kilograms / (Height in Metres2)

*This formula requires two values. You will need multiple statements get the input.*

*e.g. Someone with a weight of 80kg and a height of 1.72m has a BMI of 27.04.*

**Finance**

Interest = Amount \* (Annual Interest Rate / 100) \* Time in Years

*This formula requires three values. You will need multiple statements get the input. Do not include the dollar sign or percent symbol when entering values into the prompts.*

*e.g. A $5000 loan at an annual interest rate of 6.5% for 8 years would accrue $2600 interest.*

Most of these programs will involve an additional statement or two in order to prompt for all the necessary values or do additional calculations, but the logic behind them is the same as before.

Again, **write** a few of different programs and **test** them to ensure they are working correctly. While it is a little bit redundant for such simple programs, **write pseudocode** and **draw flowcharts** for the programs you create. How (if at all) does their design differ from the design of previous programs?

As a final exercise, enhance the output of all of the programs you have created by using the built-in method Math.**round()**. This method expect you to give it two parameters – a number to round, and the number of digits to round it to. It returns the number, rounded to the number of digits.

*That’s all for this workshop. If you haven’t completed the workshop or readings,*

*find time to do so before next week’s class*