

Tutorial Two

1. Write a program that takes the Cartesian components of two vectors, each as a **array** of 3 elements, and passes these values to **functions** that return:
 - a. the dot product,
 - b. the vector product,
 - c. the angle (in both degrees and radians) between the two vectors.
2. Pointers and references. The aim of the program **conversion.cpp** is to convert integer and floating point temperatures from Celsius to Fahrenheit by passing the values to functions as pointers or references. Implement the given function prototypes to perform the conversions and display the result of converting both 22 °C and 22.5 °C to Fahrenheit.
3. Stats functions. The aim of the program **correlation.cpp** is to find the correlation between two data samples. One data sample has a list of heights and other a list of weights. The comments throughout the code indicate the steps needed to calculate the correlation between the two data samples. Follow these steps and complete the program. Ensure that the correlation between the data sets is approximately 0.78.
4. Factors.
 - a. Write a program containing a **function** that displays the factors of a number given by the user of the program.
 - b. Adapt this function so that when called, the function will either display all the factors of the given number or just display whether the number is prime. Ensure that this choice is made by the user outside of this function.
 - c. Add a new **function** to the program that will only display the prime factors of the given number.

Hints for Question 1:

- The equations for this problem should be readily available in any maths or physics textbook.
- If you really don't know how to proceed with this question, start by writing a program to calculate two fixed value vectors and then see how the code can be modified to solve the general case.

Hints for Question 2:

- The key to answering this question is to remember the different meanings of the `*` and `&` operators.

Hints for Question 3:

- In this program you will use the functions you developed in the **stats** program. Remember that the three functions are in every way like mathematical functions **exp()** and **sqrt()**. All you need to provide is the correct number of input arguments of the correct type.
- Unlike previous uses of the **stats** program the data will be read from a file. The code for this step is already completed. Just make sure that the file **statsfile.cpp** is included in the compilation step.
- Refer to any good statistics book for the equation for a correlation coefficient!

Hints for Question 4:

- You already have the algorithm for determining factors of a given number in Lecture Two.
- There are many ways of answering b). The minimum requirement is that the function prototype is extended to accept a second argument that determines the level of output to the user. How this is implemented in the function is up to you.
- c) requires a bit of thinking! The key test is to find out if a factor of the input number is prime or not. In essence, you already have this algorithm in the code. How can this algorithm be applied again?