This document is part of the HTML publication "**An Introduction to the Imperative Part of C++**"

The original version was produced by **Rob Miller** at **Imperial College London**, September 1996.

Version 1.1 (modified by **David Clark** at **Imperial College London**, September 1997)

Version 1.2 (modified by **Bob White** at **Imperial College London**, September 1998)

Version 1.3, 1.4, 2.0, ..., 2.15 (modified by **William Knottenbelt** at **Imperial College London**, September 1999-September 2016)

Introduction to C++ Programming: Exercise Sheet 3

Question 1

Add 6 appropriate function declarations for

```
celsius_of                    absolute_value_of

print_preliminary_message     input_table_specifications

print_message_echoing_input   print_table
```

to the following program, and add the 4 missing function definitions, so that it produces output such as that given below (you may find it helpful to cut, paste and modify the code for the answer to Question 2, Exercise Sheet 2). Test your program with various inputs.

```
/* This program prints out a conversion table of temperatures, after
prompting the user for upper and lower bounds of the table in
Fahrenheit, and the temperature difference between table entries. */

#include <iostream>
using namespace std;

/* START OF MAIN PROGRAM */
int main()
{
        int lower = 0; /* the lowest Fahrenheit entry in the table */
        int upper = 0; /* the highest Fahrenheit entry in the table */
        int step = 1;  /* difference in Fahrenheit between entries */

        /* print a message explaining what the program does: */
        print_preliminary_message();

        /* prompt the user for table specifications in Fahrenheit: */
        input_table_specifications(lower, upper, step);

        /* print appropriate message including an echo of the input: */
        print_message_echoing_input(lower, upper, step);

        /* Print the table (including the column headings): */
        print_table(lower, upper, step);

        return 0;
}
```

```
/* END OF MAIN PROGRAM */

/* FUNCTION TO CONVERT FAHRENHEIT TO CELSIUS */
double celsius_of(int fahr)
{
        return (static_cast<double>(5)/9) * (fahr - 32);
}
/* END OF FUNCTION */

/* FUNCTION TO CONVERT FAHRENHEIT TO ABSOLUTE VALUE */
double absolute_value_of(int fahr)
{
        return ((static_cast<double>(5)/9) * (fahr - 32)) + 273.15;
}
/* END OF FUNCTION */
```

Program Ex3.1

Example output:

```
This program prints out a conversion table of temperatures.

Enter the minimum (whole number) temperature
you want in the table, in Fahrenheit: 0
Enter the maximum temperature you want in the table: 100
Enter the temperature difference you want between table entries: 20

Tempertature conversion table from 0 Fahrenheit
to 100 Fahrenheit, in steps of 20 Fahrenheit:

Fahrenheit       Celsius      Absolute Value

   0              -17.78         255.37
   20             -6.67          266.48
   40             4.44           277.59
   60             15.56          288.71
   80             26.67          299.82
   100            37.78          310.93
```

(EXAMPLE ANSWER) (BACK TO COURSE CONTENTS)

Question 2

Split your answer program to Question 1 into three files: (i) a main program file, (ii) a header file called "conversions.h" for the functions "`celsius_of(...)`" and "`absolute_value_of(...)`", and (iii) an implementation file for these two functions. Again, test your program with various inputs.

(EXAMPLE ANSWER: main program file, header file, implementation file) (BACK TO COURSE CONTENTS)

Question 3

(a) Create a header file "statistics.h" and a corresponding implementation file "statistics.cpp" containing the functions "`average(...)`" and "`standard_deviation(...)`". The functions should return the average and standard deviation respectively of 1, 2, 3 or 4 real number values. The standard deviation of the numbers r1, ..., rN is defined as the square root of the average of the expressions

((r1 - a) x (r1 - a)) , ((r2-a) x (r2-a)), ..., ((rN - a) x (rN - a))

where a is the average value of r1, ..., rN.

Hints:
(i) You should take advantage of C++'s facility for polymorphic functions, and overload the function names.
(ii) It is possible to call one function from inside another.
(iii) You should be doing plenty of cutting and pasting (not usually desirable!)

(b) Write a program which tests the functions in "statistics.h" again and again with various inputs until you tell the program that you are finished. Your program should be able to reproduce the following sample input/output session:

```
This program tests the functions in the 'statistics.h' header file.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 3

Enter first value: 5
Enter second value: 7
Enter third value: 9
Average: 7. Standard deviation: 1.63299.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 1

Enter first value: 5.8
Average: 5.8. Standard deviation: 0.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 8

Sorry, the program can only test 1, 2, 3 or 4 values.

Do you wish to test 1, 2, 3 or 4 numbers (enter 0 to end the program): 0

Finished testing 'statistics.h' header file.
```

Hints:
(i) Design your program "top down". Begin by writing a short main program which calls functions such as "test_three_values()", which you can define in detail later, after "main".
(ii) As a top level control structure, you might want to use a "for loop" with empty initialisation and update statements (this is equivalent to a "while statement").

(EXAMPLE ANSWER: main program file, statistics.h, statistics.cpp) (BACK TO COURSE CONTENTS)
