

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 5

### Question 1

Write a logical (i.e. Boolean) valued function which takes a single integer parameter, and returns "True" if and only if the integer is a prime number between 1 and 1000 (note that 1 is not usually counted as a prime number). Test your function by calling it from an appropriate "driver" program with different inputs.

#### Hints:

(i) if a number is not prime, it has at least one prime factor less than or equal to its square root,

(ii)  $(32 \times 32) = 1024$  and  $1024 > 1000$ .

([EXAMPLE ANSWER](#)) ([BACK TO COURSE CONTENTS](#))

### Question 2

Write a function "print\_pyramid(...)" which takes a single integer argument "height" and displays a "pyramid" of this height made up of of "\*" characters on the screen. Test the function with a simple "driver" program, which should be able to reproduce the following example output:

This program prints a 'pyramid' shape of  
a specified height on the screen.

how high would you like the pyramid?: 37  
Pick another height (must be between 1 and 30): 6

```

      **
     ****
    *
  
```

([EXAMPLE ANSWER](#)) ([BACK TO COURSE CONTENTS](#))

### Question 3

"For" loops can always be re-written as "while" loops, and vice-versa. Are the following two programs equivalent, and what is their output? Explain your answer, and run the programs to check.

[Program \(a\):](#)

```

#include <iostream>
using namespace std;

int main()
{
    int count = 1;
    for (; count <= 5 ; count++)
    {
        int count = 1;
        cout << count << "\n";
    }
}

```

```

    }
    return 0;
}

```

#### Program (b):

```

#include <iostream>
using namespace std;

int main()
{
    int count = 1;
    while (count <= 5)
    {
        int count = 1;
        cout << count << "\n";
        count++;
    }
    return 0;
}

```

([Program \(a\) output](#)) ([Program \(b\) output](#)) ([BACK TO COURSE CONTENTS](#))

## Question 4

The [following program](#) prints out a table of shop closing times. It uses a named enumeration to define a new data type "Day", and a function "closing\_time(...)" to generate the closing time from a given day (not a complicated function - the shop closes at 5pm every day). The program illustrates how the identifiers "int" and "Day" can both be used as type casts (in the head of the "for" loop and in the call to "closing\_time(...)" respectively):

```

#include <iostream>
using namespace std;

enum Day {Sun, Mon, Tue, Wed, Thu, Fri, Sat};

int closing_time(Day day_of_the_week);

/* MAIN PROGRAM */
int main()
{
    int count;

    cout.setf(ios::left);
    /* Print table heading */
    cout.width(17);
    cout << "DAY";
    cout << "CLOSING TIME\n\n";

    /* Print table from Sunday to Saturday */
    for (count = static_cast<int>(Sun) ; count <= static_cast<int>(Sat) ; count++)
    {
        cout.width(19);
        switch (count)
        {
            case Sun: cout << "Sunday"; break;
            case Mon: cout << "Monday"; break;
            case Tue: cout << "Tuesday"; break;
            case Wed: cout << "Wednesday"; break;
            case Thu: cout << "Thursday"; break;
            case Fri: cout << "Friday"; break;
            case Sat: cout << "Saturday"; break;
            default:   cout << "ERROR!";
        }
        cout << closing_time(static_cast<Day>(count)) << "pm\n";
    }

    return 0;
}

```

```

/* END OF MAIN PROGRAM */

/* FUNCTION TO GENERATE SHOP CLOSING TIMES FROM DAY OF THE WEEK */
int closing_time(Day day_of_the_week)
{
    return 5;
}

```

(a) What happens (and why) if we try to replace the "switch" statement with the single line

```
cout << static_cast<Day>(count);      ?
```

Replace the "switch" statement with the line

```
print_day(static_cast<Day>(count), cout);
```

instead, and add an appropriate definition after the main program for the function "print\_day(...)", using a "switch" statement (don't forget the function declaration as well). ("cout" is of type or class "ostream").

(b) The shop keeper wants to change the closing times to 1pm on Sundays, 5pm on Saturdays, 8pm on Wednesdays, and 6pm all other days. Make appropriate changes to the function "closing\_time(...)" using a "switch" statement, and test the program.

([EXAMPLE ANSWER](#)) ([BACK TO COURSE CONTENTS](#))

## Question 5

Write a program which prints a table listing the number of occurrences of the lower-case characters 'a' to 'z' in the file "Sheet5Ex5.cpp". (Save your program in a file of this name so that you can test it on itself.) Write your program as though your computer is short of memory - you should declare only one variable of type "ifstream", one variable of type "char", and two variables of type "int". The program should produce output such as the following:

CHARACTER	OCCURRENCES
a	38
b	5
c	35
d	7
e	58
f	8
g	8
h	21
i	32
j	0
k	0
l	19
m	16
n	34
o	22
p	8
q	0
r	48
s	17
t	61
u	15
v	0
w	4
x	4
y	0
z	1

([EXAMPLE ANSWER](#)) ([BACK TO COURSE CONTENTS](#))