

CA146 Introduction to Programming in C++

CA146 Tutorial 4

Note: As usual, when you have finished the tasks, or at least as many as you were able to do, ask a tutor to sign off to confirm this.

1 Getting started with while loops

- 1.1 Write a program that uses a while loop to print out the numbers 0 through 99.
- 1.2 Write a program that uses a while loop to read in 5 numbers and calculate their sum (you will need a sum variable initialised to zero to which on each iteration of the loop you add the latest number entered)
- 1.3 The Fibonacci sequence of numbers is given by: 1, 1, 2, 3, 5, 8, 13... The first two numbers of the sequence are both defined to be 1 and thereafter each number in the sequence is defined as the sum of the previous two. Write a program that uses a while loop to display the first 20 numbers of the Fibonacci sequence.

CA146 Introduction to Programming in C++

2 Practice with single loop and decisions

Suppose that the following polynomial represents the altitude of a weather balloon in metres during the first 48 hours after its launch, where “t” is in hours:

$$\text{alt}(t) = -0.12t^4 + 12t^3 - 380t^2 + 4100t + 220$$

and that, correspondingly, its velocity in metres per hour is given by

$$v(t) = -0.48t^3 + 36t^2 - 760t + 4100.$$

The following program prints a table of values of the altitude and the velocity using units of metres and metres/second. It requests the user to enter, in hours, the starting time, the increment in time between lines of the table, and the ending time; all the time values should be less than 48 hours.

2.1 Type in, compile and run this program.

```
#include <iostream>
using namespace std;

int
main()
{
    float initial, increment, final;
    float t, altitude, velocity;

    // Read in table parameters:
    cout << "Enter start time, increment & finish time values (in hours):" << endl;
    cin >> initial >> increment >> final;

    // Print table headings
    cout << "Weather Balloon Information (within first 48 hours)" << endl;
    cout << "Time\tAltitude\tVelocity" << endl;
    cout << "(h)\t(m)\t(m/s)" << endl;

    // Compute and print report
    t = initial;
    while (t <= final) {

        altitude = (-0.12 * t * t * t * t) + (12.0 * t * t * t)
            - (380.0 * t * t) + (4100.0 * t) + 220.0;

        velocity = (-0.48 * t * t * t) + (36.0 * t * t) - (760.0 * t) + 4100.0;

        // Display
        cout << t << "\t" << altitude << "\t" << velocity / 3600.0 << endl;

        t = t + increment;
    }
    return (0);
}
```

CA146 Introduction to Programming in C++

NB: As usual, check the results carefully to make sure you understand exactly how the program works to produce the different outputs. (The table may look a bit untidy but this can be dealt with later on.)

2.2 Modify the program of 2.1 so that it also finds and prints out the maximum altitude reached as well as the corresponding time when this occurred. Hints:

- (a) Introduce two new “float” variables, called tMax and altitudeMax (say). By the end of execution, altitudeMax should store the maximum altitude and tMax the corresponding time.
- (b) Before the “while” statement initialise both variables by including “tMax=initial; altitudeMax=0;”
- (c) You will need to include an “if statement” just before “t=t+increment;” the purpose of which is to update the values of altitudeMax and tMax if necessary (i.e. if altitude > altitudeMax).
- (d) Finally, just before “return (0);” you need to output the values of altitudeMax and tMax. Verify your program prints the correct answers by examining the table yourself.

CA146 Introduction to Programming in C++

3 Practice with nested “while” loops

3.1 Type in, compile and run the following program, the purpose of which is to calculate the pressure of a container using the equation of state ($PV = nRT$) for an ideal gas. As written the program is restricted to a constant value of $n (= 1)$ and to a container of volume 0.1 m^3 .

```
#include <iostream>
using namespace std;

int
main()
{
    float tempC; // Degrees C
    float tempK; // Degree K
    float volume; // metre cubed
    float pressure; // n per metre squared
    int n; // Number of moles
    float R = 8.3; // Constant in J/mol K

    n = 1; // Fix the value of n (rather restrictive)
    volume = 0.1; // Fix the value of volume (rather restrictive)

// **

    cout << "Pressure as a function of temperature for V = "
         << volume << "m^3" << endl;

    cout << "P(n/m^2)\tT(K)" << endl;

    tempC = 0;

    while (tempC <= 100.0) {

        tempK = tempC + 273.0;
        pressure = n * R * tempK / volume;
        cout << pressure << "\t\t" << tempK << endl;
        tempC = tempC + 10.0;
    }
    return (0);
}
```

3.2 Extend the above program so that it produces a **series** of tables for containers of volume 0.1m^3 , 0.2m^3 , ..., 0.5m^3 . (**Hint**: you will need to put in an extra (outer) “while” loop starting at the line marked **).