```cpp
/* The following code example is taken from the book
 * "The C++ Standard Library - A Tutorial and Reference, 2nd Edition"
 * by Nicolai M. Josuttis, Addison-Wesley, 2012
 *
 * (C) Copyright Nicolai M. Josuttis 2012.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include <future>
#include <thread>
#include <chrono>
#include <random>
#include <iostream>
#include <exception>
using namespace std;

int doSomething (char c)
{
    // random-number generator (use c as seed to get different sequences)
    std::default_random_engine dre(c);
    std::uniform_int_distribution<int> id(10,1000);

    // loop to print character after a random period of time
    for (int i=0; i<10; ++i) {
        this_thread::sleep_for(chrono::milliseconds(id(dre)));
        cout.put(c).flush();
    }

    return c;
}

int func1 ()
{
    return doSomething('.');
}

int func2 ()
{
    return doSomething('+');
}

int main()
{
    std::cout << "starting func1() in background"
              << " and func2() in foreground:" << std::endl;

    // start func1() asynchronously (now or later or never):
    std::future<int> result1(std::async(func1));

    int result2 = func2();    // call func2() synchronously (here and now)

    // print result (wait for func1() to finish and add its result to result2
    int result = result1.get() + result2;
    std::cout << "\nresult of func1()+func2(): " << result
              << std::endl;
}
```