```
//  file: gaussian_random.cpp
//
//  Program to illustrate how to generate uniformly distributed
//   and gaussian distributed random numbers using GSL routines
//
//  Programmer:  Dick Furnstahl  furnstahl.1@osu.edu
//
//  Revision history:
//      20-Feb-2004  original version, converted gaussian_random.c
//      19-Feb-2005  minor changes to variable names and comments
//      13-Feb-2006  tidied up the code, moved declarations
//
//  Notes:
//   * uses the GSL random number functions
//   * both the gsl_rng.h and gsl_randist.h header files are needed
//   * the current version uses the gsl_rng_taus random number
//      generator.  There are many other choices (just change
//      the name in the gsl_rng_alloc statement).  See the GSL
//      manual for a list of generators and their properties.
//   * The Gaussian random variate has mean zero and standard deviation
//      sigma.  Its probability distribution is
//        p(x)dx = 1/Sqrt[2 Pi sigma^2] Exp[-x^2/(2 sigma^2)]
//
//*****************************************************************

// include files
#include <iostream>             // cout and cin
#include <iomanip>              // manipulators like setprecision
#include <fstream>              // file input and output
using namespace std;            // we need this when .h is omitted

#include <gsl/gsl_rng.h>        // GSL random number generators
#include <gsl/gsl_randist.h>    // GSL random distributions


//*****************************************************************
int
main ()
{
  gsl_rng *rng_ptr;             // pointer to random number generator (rng)
  rng_ptr = gsl_rng_alloc (gsl_rng_taus);       // allocate the rng

  unsigned long int seed;       // "seed" for the random number generators
  cout << "Enter a long integer as a seed: ";
  cin >> seed;
  gsl_rng_set (rng_ptr, seed);  // seed the rng

  // output file for numbers
  ofstream out;
  out.open ("random_numbers.dat");

  // generate uniform and gaussian distributed random numbers
  int npts = 100;               // number of random numbers to generate
  cout << "How many random numbers? ";
  cin >> npts;

  double lower = 0.;            // lower limit for uniform region
  double upper = 1.;            // upper limit for uniform region
  double sigma = 1.;            // standard deviation of gaussian distribution

  out << "# uniform1 uniform2 gaussian1 gaussian2" << endl;
  for (int i = 0; i < npts; i++)
    {
      // uniform random numbers from [lower,upper]
      double uniform1 = gsl_ran_flat (rng_ptr, lower, upper);
      double uniform2 = gsl_ran_flat (rng_ptr, lower, upper);

      // random numbers distributed as gaussians
      double gaussian1 = gsl_ran_gaussian (rng_ptr, sigma);
```



```
      double gaussian2 = gsl_ran_gaussian (rng_ptr, sigma);

      out << " " << fixed << uniform1 << " " << uniform2 << " "
          << gaussian1 << " " << gaussian2 << endl;
    }

  gsl_rng_free (rng_ptr);       // free the random number generator
  cout << "Output " << npts << " random numbers to random_numbers.dat." << endl;
  out.close ();                 // close the output file

  return 0;
}
```