```c
/****************************************************
 * random_number.c:    extra material for sheet 2,    *
 *                     explaining basic use of random *
 *                     variables                      *
 *                     dated 17.04.19                 *
 *****************************************************/

#include <stdio.h>
#include <time.h>       // defines time()
#include <stdlib.h> // defines rand()


/*
        This is a macro. Macros are written in capital letters
        and are replaced in the code by the compiler in the preprocessor
        compilation stage by the expression that they are followed
        up with. They are a convenient way for introducing constants.
*/
#define CONSTRAINED_INT_MAX 10

int main(){
        /*
                We first set the seed of the pseudo number generator.
                This can be done in multiple ways. An very basic method
                is to use the current system time as the initial seed,
                where time(NULL) is the system time (in number of seconds
                dating back from January 1 1970). The srand function then
                sets the seed utilised by rand().
        */
        srand( time(NULL) );
        // NULL is a variable of pointer type, used here for historic reasons.

        /*
                From the seed, pseudo random numbers are generated by the use
                of rand() where the seed is inserted into a highly nonlinear
                function, returning the output and updating the seed
                afterwards.
        */
        printf("My first random integer: %d\n", rand() );
        printf("My second random integer: %d\n", rand() );
        printf("My third random integer: %d, etc.\n", rand() );
        /*
                Note that rand() takes no arguments, the seed is defined
                as an external variable which can be accessed by all
                functions. The output is an unsigned integer not larger
                than the value of the macro RAND_MAX.
        */

        /*
                In most application, we want our random number to be in a
                certain value range. This can be achieved by manipulating
                the result of rand() using the modulo operator '%'. Here we
                compute a random number between 0 and the macro
                CONSTRAINED_INT_MAX.
        */
        int constrained_random_int = rand() % (CONSTRAINED_INT_MAX + 1);
        printf("Constrained random integer: %d\n", constrained_random_int);

        /*
                If we want to set random values to a variable, which is not an
                integer type, we need to use a type cast. For instance, in
                order to obtain a random double in the range between 0  and 1,
                we can proceed as follows
        */
        double random_double = ( (double) rand() )/RAND_MAX;
        /*
                The explicit type cast is important, as otherwise integer
                division is performed and the result would be equal to
                zero always!
	*/
	printf("Constrained random double: %.2f\n", random_double);

	/*
		This procedure can easily be extended to all other arithmetic
		types (such as chars ;) ) and to various random value ranges
		by the use of offsets and multipliers.
	*/

	return 0;
}
```