

# Random Numbers in Traditional C++

## Exercises

- What does rand() do?
  - rand() calls a pseudo-random number generator and returns the next number in the sequence
- What does RAND\_MAX signify?
  - The maximum value rand() can return
- Write a simple program which uses rand() to display a random number
- Run the program again. What do you notice? Same result

- Write a simple program which uses `rand()` to
  - a) Print out a floating point number with a random value between 0 and 1
  - b) Print out ten integers with random values between 0 and 100

- Give some disadvantages of using rand()
  - Some implementations provide poor quality random numbers
  - We often need to rescale the range (e.g. to obtain random numbers between 0 and 1)
    - This introduces “bias” in which some numbers appear more often than others
    - Also programming errors(!)
  - Not suitable for secure applications as the sequence repeats too frequently

- Explain how to seed the generator used by rand()
  - Call srand()
- What is a popular technique for doing this?
  - srand(time(0)); // Uses current time as seed

- Alter your program from the last exercise so it seeds the generator in this way
- Run the program again. What do you notice?
  - The sequence of numbers is different each time the program is run
- Give a drawback of the seeding technique you used
  - time(0) returns the current time to the nearest second
  - If two PRNGs are started at the same time, they will use the same seed and generate the same sequence of numbers