

ELEC0021 - PROGRAMMING II OBJECT-ORIENTED PROGRAMMING

RANDOM NUMBER GENERATION

Prof. George Pavlou
Communication and Information Systems Group
Dept. of Electronic and Electrical Engineering
<http://www.ee.ucl.ac.uk/~gpavlou/>
g.pavlou@ucl.ac.uk

Random Numbers

- **Random numbers** emulate the random occurrence of events and are used extensively for simulation and game playing related programming
- Random numbers in Java can be produced through methods of class **Random** (package `java.util.Random`) or through the static method **random** of class `Math`
 - Objects of class `Random` can produce random boolean, int, long, float, double and Gaussian values
 - `Math.random` can produce only double values in the range $0.0 \leq x < 1.0$
- They are effectively pseudo-random as a series is produced through a complex mathematical calculation i.e. for a given “seed”, the derived series is deterministic
 - A seed is needed to start a series; the current time is typically used to introduce series randomness
 - An explicit seed can be provided for result reproducibility

Random class *nextInt* method

- The **Random** class **nextInt** method produces random int values in the range $-2^{31} \leq x \leq 2^{31}-1$ (Java integers are 32 bit entities)
 - `Random randGen = new Random();`
`int randomInt = randGen.nextInt();`
- **nextInt** with a positive integer argument returns random values in the range $0 \leq x < \text{argument}$
 - `nextInt(2)` emulates random boolean values (0 or 1)
 - `nextInt(6)` emulates random dice rolling, in fact
face = 1+nextInt(6) shifts the value to the actual dice face
- Creating a random generator with a specific seed guarantees repeatability by producing the same series
 - `Random randGen = new Random(seed);`

A Simple CoinThrower Program

```
import java.util.Random;

public class CoinThrower {

    public static void main (String[] args) {
        // first declare and initialise face up/down integer variables
        int faceUp = 0, faceDown = 0; // Java will initialise anyway but C/C++ will NOT
        Random randGen = new Random(); // create new random generator object

        for (int i = 1; i <= 1000; i++) {
            int thisFace = randGen.nextInt(2); // generate next random 0 or 1
            if (thisFace == 1) // nextInt(2) returned 1 i.e. up
                faceUp++;
            else                // nextInt(2) returned 0 i.e. down
                faceDown++;
        }

        System.out.println("faceUp came " + faceUp + " times");
        System.out.println("faceDown came " + faceDown + " times");
    }
}
```

Another Example Test Program

- You should implement a test program that emulates the throwing of a virtual dice (of any number of sides) a few thousand times and check if the occurrence frequency of every face is roughly the same – see assignment
- Use the CoinThrower program as a starting point

Summary

- Random numbers are very useful for game playing and simulation
 - The Random class nextInt method is commonly used to produce integer random values
- In C, the “int random(int)” function does something similar to “int Random.nextInt(int)”
 - The coin thrower program could be also done in C
 - We do not really use O-O features in this program