Random Events

In the real world, random events occur all the time. Inside the computer, random events are (thankfully) rare. For a computer, a random event usually means trouble.

In order to model real world situations, a computer is often required to simulate a random event.

Most programming languages provide some sort of random number generator to allow this.

Example of Random Events

- flipping a coin (50% heads, 50% tails)

- rolling a die (1 in 6 chance of getting each face)

- selecting spheres at a bingo hall

- game shows such as Wheel of Fortune, Deal or No Deal

Random Number Generator

Most random number generators will produce a real value between 0 and 1, as a decimal. For example: 0.1, 0.445, 0.999, 0.5

How can numbers like these be used to simulate:

- a flipped coin?
- a rolled die?

Flipping a Coin (simulation)

```
var num : real

rand(num)    % assign a random
             % value to num

if (num > 0.5) then
  put “Heads”
else
  put “Tails”
end if
```

Rolling a Die (simulation)

When we flipped our “coin”, there were only two choices for the if statement. With a die, there are 6, and the program would continue to get worse for larger values.

Fortunately, Turing has a built-in command to produce random integer values.

randint(*variable, lowerLimit, upperLimit)*

Rolling a Die (simulation)

```
% roll a single die 5 times

var roll : int

for count : 1 .. 5
  randint ( roll, 1, 6 )
  put “You rolled a “, roll
end for
```