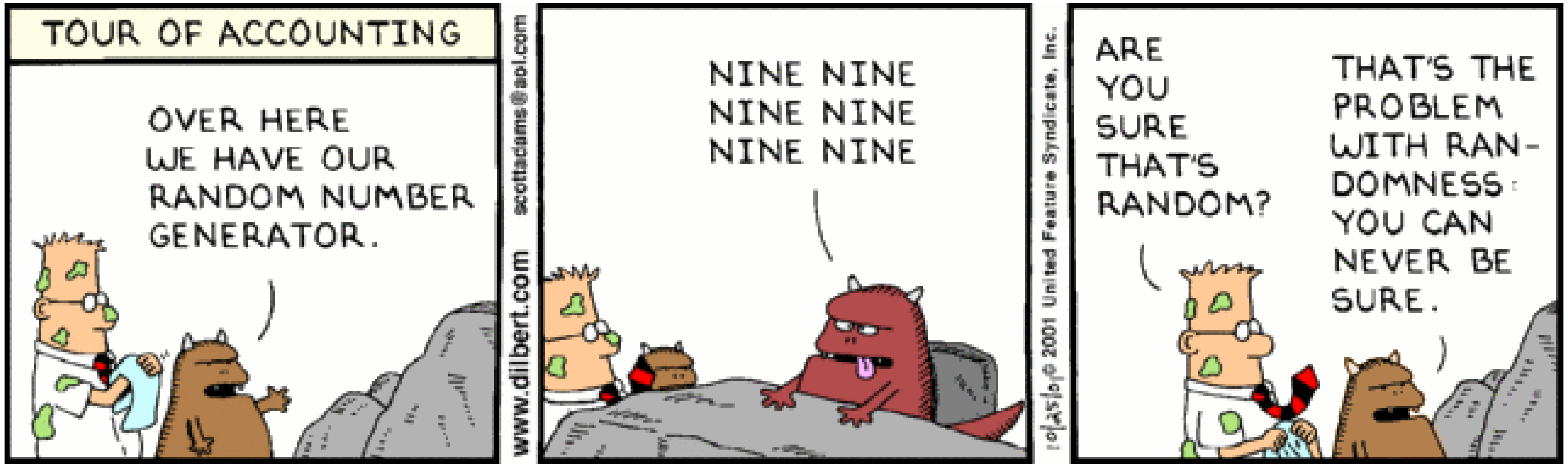
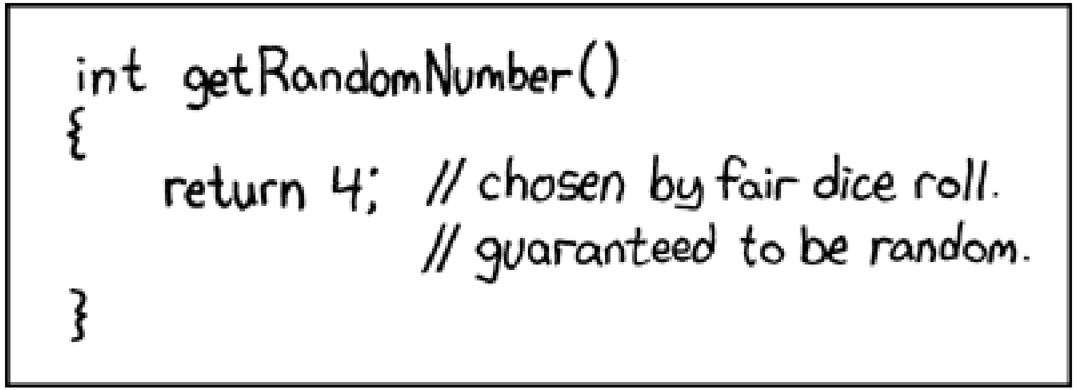
CSc 110, Autumn 2016

Lecture 13: Random Numbers

Adapted from slides by Marty Stepp and Stuart Reges


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
www.dilbert.com scottadams@aol.com
NINE NINE NINE NINE NINE NINE
10/25/01 © 2001 United Feature Syndicate, Inc.
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.



```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

http://xkcd.com/221/

Randomness

- Lack of predictability: don't know what's coming next

- Random process: outcomes do not follow a deterministic pattern (math, statistics, probability)

- Lack of bias or correlation (statistics)

- Relevant in lots of fields
    - Genetic mutations (biology)
    - Quantum processes (physics)
    - Random walk hypothesis (finance)
    - Cryptography (computer science)
    - Game theory (mathematics)
    - Determinism (religion)

Pseudo-Randomness

- Computers generate numbers in a predictable way using a mathematical formula
- Parameters may include current time, mouse position
  - In practice, hard to predict or replicate
- True randomness uses natural processes
  - Atmospheric noise (http://www.random.org/)
  - Lava lamps (patent #5732138)
  - Radioactive decay

The Random class

- A random object generates pseudo-random numbers.
  - Class random is found in random

    ```
    from random import *
    ```

Method name	**Description**
`random()`	returns a random float in the range [0, *1*) in other words, 0 inclusive to *max* exclusive
`randint(`***min, max***`)`	returns a random integer in the range [min, *max*) in other words, min to *max*-1 inclusive

  - Example:

```
from random import *
random_number = randint(1, 10)    # 1-9
```

Generating random numbers

- To get a number in arbitrary range [*min*, *max*] inclusive:

```
randint(min, max)
```

  - Where ***size of range*** is (***max*** – ***min***)

- Example: A random integer between 4 and 10 inclusive:

```
n = randint(4, 11)
```

Random and other types

- `random` function returns a `float` between 0.0 - 1.0
  - Example: Get a random GPA value between 1.5 and 4.0:

```
random_gpa = random() * 2.5 + 1.5
```

- Any set of possible values can be mapped to integers
  - code to randomly play Rock-Paper-Scissors:

```
r = randint(0, 3)
if (r == 0):
    print("Rock")
elif (r == 1):
    print("Paper")
else:   # r == 2
    print("Scissors")
```

Random question

- Write a program that simulates rolling two 6-sided dice until their combined result comes up as 7.

```
2 + 4 = 6
3 + 5 = 8
5 + 6 = 11
1 + 1 = 2
4 + 3 = 7
You won after 5 tries!
```

Random answer

```
# Rolls two dice until a sum of 7 is reached.
From random import *

def main():
    tries = 0

    sum = 0
    while (sum != 7):
        # roll the dice once
        roll1 = randint(1, 7)
        roll2 = randint(1, 7)
        sum = roll1 + roll2
        print(str(roll1) + " + " + str(roll2) + " = " + str(sum))
        tries += 1

    print("You won after " + str(tries) + " tries!")
```

Random question

- Write a program that plays an adding game.
  - Ask user to solve random adding problems with 2-5 numbers.
  - The user gets 1 point for a correct answer, 0 for incorrect.
  - The program stops after 3 incorrect answers.

```
4 + 10 + 3 + 10 = 27
9 + 2 = 11
8 + 6 + 7 + 9 = 25
Wrong! The answer was 30
5 + 9 = 13
Wrong! The answer was 14
4 + 9 + 9 = 22
3 + 1 + 7 + 2 = 13
4 + 2 + 10 + 9 + 7 = 42
Wrong! The answer was 32
You earned 4 total points
```

Random answer

```
# Asks the user to do adding problems and scores them.
from random import *

def main():
    # play until user gets 3 wrong
    points = 0
    wrong = 0
    while (wrong < 3):
        result = play()      # play one game
        if (result == 0):
            wrong += 1
        else:
            points += 1

    print("You earned " + str(points) + " total points.")
```

Random answer 2

```
# Builds one addition problem and presents it to the user.
# Returns 1 point if you get it right, 0 if wrong.
def play():
    # print the operands being added, and sum them
    operands = randint(2, 6)
    sum = randint(1, 11)
    print(sum, end='')

    for i in range(2, operands + 1):
        n = randint(1, 11)
        sum += n
        print(" + " + str(n), end='')
    print(" = ", end='')

    # read user's guess and report whether it was correct
    guess = input()
    if (guess == sum):
        return 1
    else:
        print("Wrong! The answer was " + str(total))
        return 0
```