Generate Random Number

Section 4.10

Method to Generate a Random Int

int rand (void);

- Returns a pseudo-random integral number in the range between 0 and RAND_MAX. RAND_MAX is defined in <cstdlib>.

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
    //print out 3 random ints
    for (int i = 0; i < 3; i++)
        cout << rand() << endl;

    return 0;
}
```

Method srand

void srand (unsigned int seed);

- **Initialize random number generator**

A sequence of random integers are generated from that seed.

- In order to generate random-like numbers, srand is usually initialized to some distinctive runtime value, like the value returned by function time (declared in header <ctime>).

```
/* initialize random seed: */
srand (time(NULL)); //time(NULL) gets the current time
```

Use srand and rand together

- To see different random integers in different running time, call srand(time(NULL)); once before calling rand().

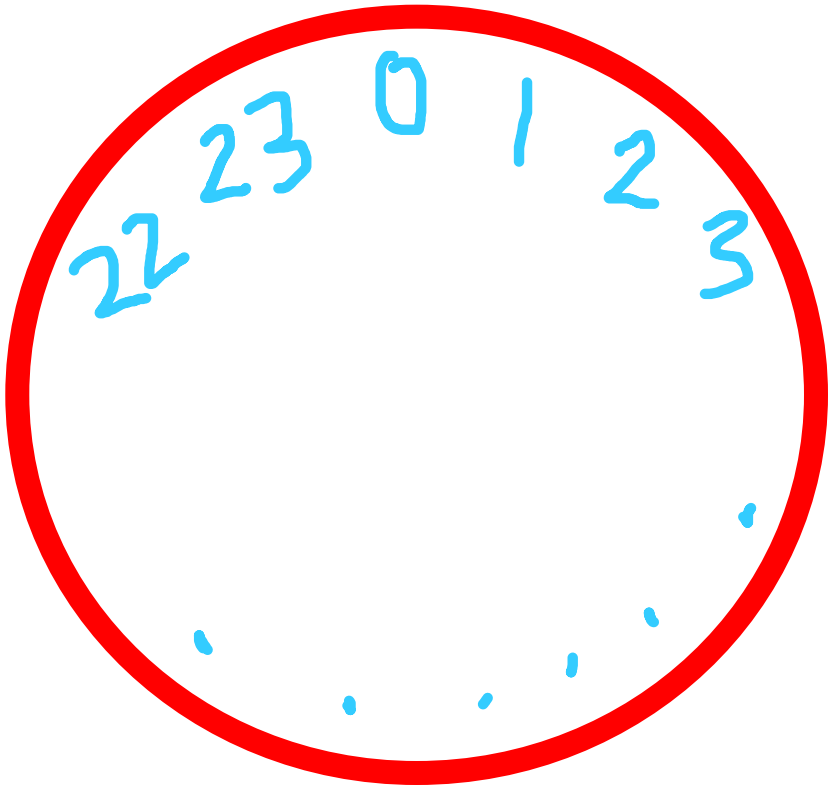
```
#include <iostream>
#include <cstdlib> //srand, rand, RAND_MAX
#include <ctime> //time
using namespace std;

int main()
{
    srand(time(NULL)); //same as srand(time(0));
    //print out 3 random ints
    for (int i = 0; i < 3; i++)
        cout << rand() << endl;

    return 0;
}
```

Throw a dice = generate random int in [1, 6]

- Hint: time is infinite, but how to label time by hours as military time?
- How many integers in [1, 6]?



Generate a random int in [1, 6]: II

- What is the result of rand() % 6?
- Given a random int in [0, 5], how to get a random int in [1, 6]?

- Image source: https://www.amazon.com/Binglinghua-Automatic-launcher-Ping-Pong-Tennis/dp/B07BBH3Z32

Generate a random int in [start, end]

- How many integers in [start, end]?
- How to generate a random int in [0, end – start]?
- How to generate a random int in [start, end]?

Easier and better random generator in C++11

- **Random** (see https://www.cplusplus.com/reference/random/)

This header introduces random number generation facilities.
This library allows to produce random numbers using combinations of *generators* and *distributions*:

- **Generators:** Objects that generate uniformly distributed numbers.
- **Distributions:** Objects that transform sequences of numbers generated by a generator into sequences of numbers that follow a specific random variable distribution, such as uniform, Normal or Binomial.

```
std::default_random_engine generator;
std::uniform_int_distribution<int> distribution(1,6);
int dice_roll = distribution(generator);  // generates number in the range 1..6
```

Generate a random floating point number in [start, end]

- Generate a random floating point number in [0, 1].

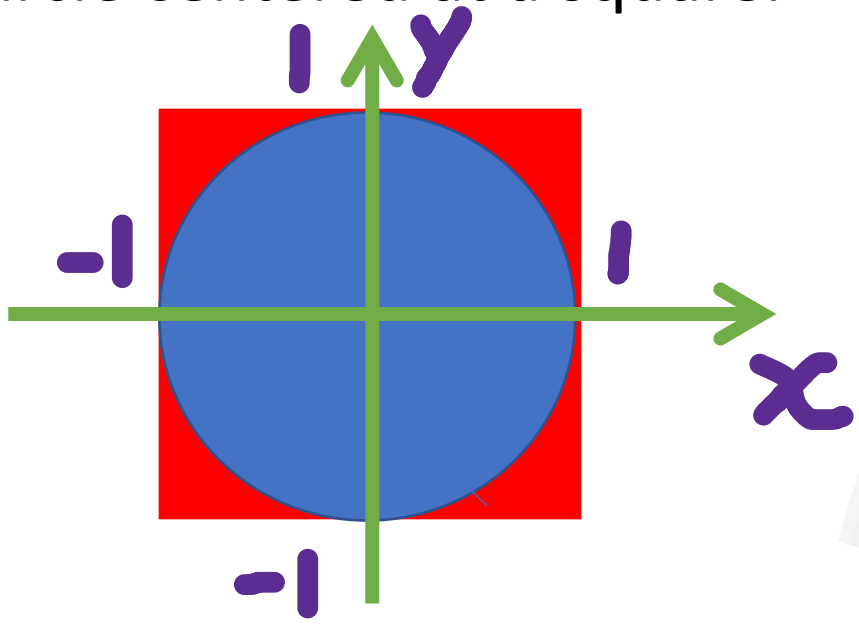
- Generate a random floating point number in [0, end – start]. It is like to draw a line in map by scaling up (or scale down if end-start <1)

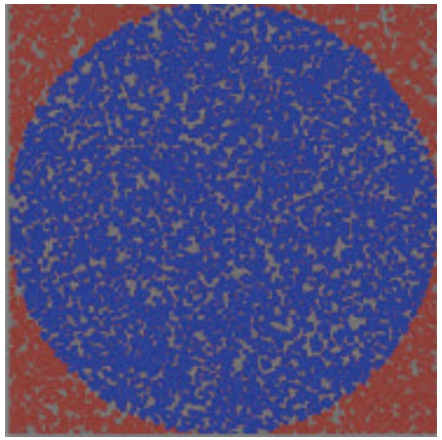
image source: https://www.cbj.ca/how-to-stretch-the-elastic-band/

- Generate a random floating point number in [start, end].

Use Monte Carlo simulation to estimate PI

- Given a unit circle centered at a square.

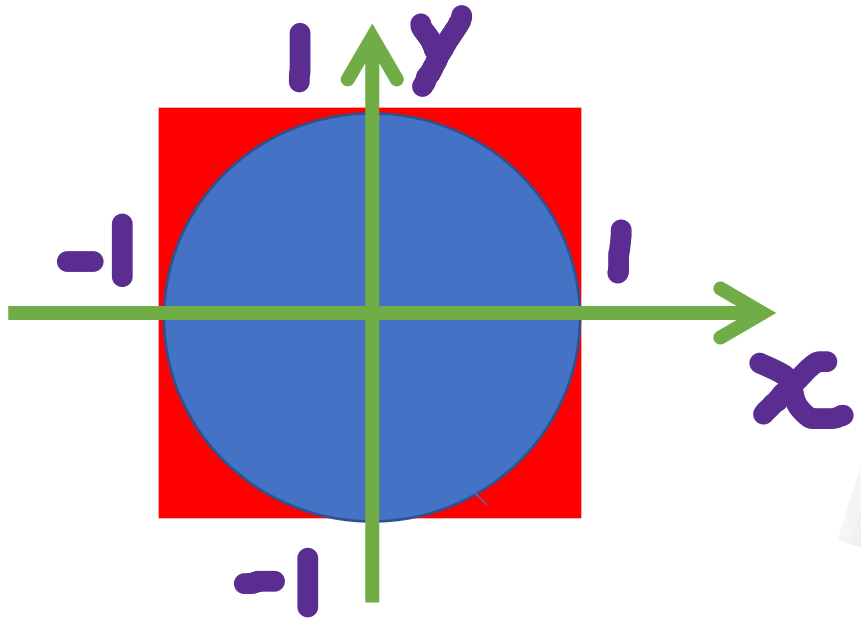


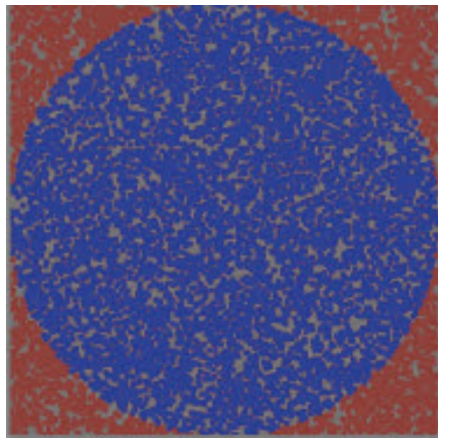


- What is the area of circle? What is the area of the square?

Use Monte Carlo simulation to estimate PI: II

- Throw a dart many times (say, 10000 or more) at the following picture, where x and y are independent random floating point number in [-1, 1].





- In the right-hand picture, a dot is like the mark left by the dart.
- What is the chance (aka probability) that the dart falls in the circle?
- How to estimate the probability?

Pseudo-code of Monte Carlo Simulation of PI

Set TRIES to be some sufficient big int, say, 10000.

Let numHits to be the number of times that the dart hits the circle, initialize it to be zero.

For each try

Begin

    Generate a random floating point number x in [-1, 1].

    Generate a random floating point number y in [-1, 1].

    if (x, y) falls inside or on the edge of the circle

        increase numHits by 1

End

Estimate PI by numHits and TRIES.

Code of using Monte Carlo to estimate pi

```
//Use Monte Carlo method to estimate pi, code link
#include <iostream>
#include <cstdlib> //rand, srand, RAND_MAX
#include <ctime>
using namespace std;

int main() {
    int numHits = 0;
    const int TRIES = 10000;
    srand(time(NULL));
    double x, y, r;
```

Using Monte Carlo to estimate pi

```
for (int i = 0; i < TRIES; i++) {
        r = 1.0 * rand() / RAND_MAX; //[0, 1]
        //r = static_cast<double>(rand()) / RAND_MAX;
            //also ok
        x = 2 * r -1; //[-1, 1]
        r = 1.0 * rand() / RAND_MAX;
        y = 2 * r -1; //[-1, 1]
        if (x * x + y * y <= 1)
            numHits++;
}
```

Using Monte Carlo to estimate pi

```
    double pi = static_cast<double>(numHits) /
TRIES * 4;
    cout << "estimated pi is " << pi << endl;

    return 0;
}
```