

Lecture 6

Random Numbers and Using cmath

Instructor: Ashley Gannon

ISC3313 Fall 2021



Random Numbers



Rand class

Package

<stdlib.h>

Usage example

```
int firstRandomNumber = rand();
```

Reference

<http://www.cplusplus.com/reference/cstdlib/rand/>



Using the Rand class

A random number between 0 and 10?

```
int r = rand()%11; //Count the numbers in [0, 10] - there are 11
```

A random number between 1 and 10?

```
int r = rand()%10 + 1; //Count the numbers in [1, 10] - there are 10. +1 starts from 1.
```

A random number between -10 and 10?

```
int r = rand()%21 - 10; //Count the numbers in [-10, 10] there are 21. -10 starts from -10.
```

A random number between 0 and 1? (a decimal number)

```
float r = rand() / RAND_MAX; //RAND_MAX is the maximum value of rand()
```



Using the Rand class

A random number between 0 and 10?

```
int r = rand()%11; //Count the numbers in [0, 10] - there are 11
```

A random number between 1 and 10?

```
int r = rand()%10 + 1; //Count the numbers in [1, 10] - there are 10. +1 starts from 1.
```

A random number between -10 and 10?

```
int r = rand()%21 - 10; //Count the numbers in [-10, 10] there are 21. -10 starts from -10.
```

A random number between 0 and 1? (a decimal number)

```
float r = rand() / RAND_MAX; //RAND_MAX is the maximum value of rand()
```

If we compile the same code multiple times, we see that we get the same random number over and over again.



Using the Rand class

- It is not enough to only use the `rand()` function to make the C++ generate random numbers.
- If you do not use the `srand` method together with `rand`, you will get the same sequence every time code runs.
- To avoid the repetitive sequence, you must set the seed as an argument to the `srand()` method. However, setting a fixed value for the `srand()` is also not a good option as the output remains the same.
- A very useful tip to make C++ generate random numbers is to use the `time()` method. By seeding the generator with the same number, you are more likely to get the same random number each time.
- Therefore, if you use the C++ `srand()` with the current time, the generated random number will always be different.



Time class

Package

`<ctime>`

Seeding Rand()

```
srand((unsigned) time(0));
```

A random number between 0 and 10?

```
srand((unsigned) time(0));  
int r = rand()%11;           //Count the numbers in [0, 10] - there are 11
```

You can try this with the other examples for practice.



cmath Basics



Commands in cmath

Package

<cmath>

Trigonometric functions

<code>cos</code>	Compute cosine (function)
<code>sin</code>	Compute sine (function)
<code>tan</code>	Compute tangent (function)
<code>acos</code>	Compute arc cosine (function)
<code>asin</code>	Compute arc sine (function)
<code>atan</code>	Compute arc tangent (function)
<code>atan2</code>	Compute arc tangent with two parameters (function)

Hyperbolic functions

<code>cosh</code>	Compute hyperbolic cosine (function)
<code>sinh</code>	Compute hyperbolic sine (function)
<code>tanh</code>	Compute hyperbolic tangent (function)
<code>acosh</code> <small>C++11</small>	Compute area hyperbolic cosine (function)
<code>asinh</code> <small>C++11</small>	Compute area hyperbolic sine (function)
<code>atanh</code> <small>C++11</small>	Compute area hyperbolic tangent (function)



Commands in math.h

Exponential and logarithmic functions

exp	Compute exponential function (function)
frexp	Get significand and exponent (function)
ldexp	Generate value from significand and exponent (function)
log	Compute natural logarithm (function)
log10	Compute common logarithm (function)
modf	Break into fractional and integral parts (function)
exp2 <small>C++B</small>	Compute binary exponential function (function)
expm1 <small>C++B</small>	Compute exponential minus one (function)
ilogb <small>C++B</small>	Integer binary logarithm (function)
log1p <small>C++B</small>	Compute logarithm plus one (function)
log2 <small>C++B</small>	Compute binary logarithm (function)
logb <small>C++B</small>	Compute floating-point base logarithm (function)
scalbn <small>C++B</small>	Scale significand using floating-point base exponent (function)
scalbln <small>C++B</small>	Scale significand using floating-point base exponent (long) (function)

Power functions

pow	Raise to power (function)
sqrt	Compute square root (function)
cbrt <small>C++B</small>	Compute cubic root (function)
hypot <small>C++B</small>	Compute hypotenuse (function)

There are many more! Check out <http://www.cplusplus.com/reference/cmath/>



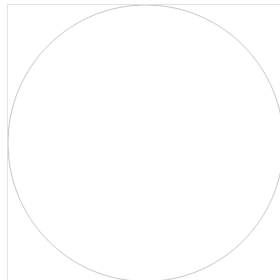
Class activity



Calculating pi

In today's activity we are going to write a C++ program to estimate the value of pi using a randomized algorithm. Imagine that we have a box with dimensions $(0, 1) \times (0, 1)$, and within it is a circle of radius 0.5. The areas are then

$$A_{circle} = \pi r^2 = \pi/4$$
$$A_{square} = 1 \times 1 = 1.$$



Calculating pi

Let's imagine that we can throw darts in this domain at random. The ratio of the darts that land inside the circle to the total should be the same as the ratio of the area of the circle to the area of the square. Given coordinates of the 'dart' (x, y) , we know it lands inside the circle if

$$\sqrt{(x - 0.5)^2 + (y - 0.5)^2} < 0.5$$



Calculating pi

The number of darts that satisfy this criteria are denoted N_{hit} . Then the ratios are equated as

$$\frac{A_{circle}}{A_{square}} = \pi \left(\frac{1}{2}\right)^2 = \frac{\pi}{4} \approx \frac{N_{hit}}{N_{tot}}$$

Then we can compute pi as

$$\pi \approx 4 \frac{N_{hit}}{N_{tot}}$$



Programming hints

Please take note of the following:

- need to be aware that two integers cannot be used in division with the expectation of getting a decimal number - Reflect on our `Types.cpp` code we wrote in class.
- must use type-conversion - if we have `int a` and `int b`, convert to double using `(double) a (double) b`)
- we can use `#define` to define constant parameters - you cannot change constant parameters.
- we have to seed the random number generator with the line `srand(time(NULL))`
- to get a random number between 0 and 1 use `((double) rand() / (RAND_MAX))`



Instructions

Download the file `computepi.cpp` from Canvas and fill in the missing code needed to approximate the value π .

