Functions

Kind of like what you learned in math class:

$$f: \mathbb{R} \to \mathbb{R}$$

$\mathbb{R}$ (domain) → $\mathbb{R}$ (codomain)

In C++:

```
double f(double);
```

double: codomain; double: domain

Important differences:

① C++ functions can have side effects, e.g. printing to stdout, change global variables...

Hence, same input might not give same output:

```
int x = 0; // global var
int f(int y) { return (y*(x++)); }
int main() {
    while (true)
        cout << f(1);
}
```

this will set y = 1

```
    return 0;
}
```

output: 0 1 2 3 4 ...

c++ functions

(2) ✓ Must have a concrete description of how to effectively compute the result.

In math world, this is fine:

$$f : \mathbb{R} \to \{0,1\}$$

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is transcendental} \\ 0 & \text{else} \end{cases}$$

This just says what it does, but not how.

Quick note: Say A, B are finite sets.

How many functions are there from $A \to B$?

What is $|\{f : A \to B\}|$

Say $A = \{a_1, a_2, \ldots a_n\}$ $(|A| = n)$.

How many choices do you have for $f(a_1)$? $|B|$. Same for $f(a_2)$, $f(a_3)$ ...

$$|B|^{|A|}$$

Even something like `bool f(int x);`

has $2^{2^{32}}$ possibilities! This would take at least 2^{32} bits to write down in general!

The point: not every mathematical function — even with reasonable domain/codomain — has a short description. Most interesting functions have some reasonable mathematical description (else, how would we even talk about them?!?), but it is a very interesting problem to see which interesting functions have reasonable descriptions as programs. Wiki "complexity theory" for more...