

Week 3

EECS 183

Variables

Types of variables

bool	true, false
char	'a', 'b', '1', '!', ' ', '\n', ...
int	-1, 0, 1, 2, ...
double	-1.5, 1.0, 3.14, ...
string	"hello", "EECS 183", "a", "", ...

Escape sequences

`\n` `\t` `\'` `\"` `\\` `...`

What is the output?

```
cout << "Hi! \" // comment";
```

Operators

+ - * / % =

%

What is the value of result?

```
double result = static_cast<double>(1 / 4);
```


I/O (input/output)

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

```
int main()
{
    cout << "Please give me an int " << endl;
    int number;
    cin >> number;
    cout << "Thanks for the " << number << "!" << endl;
}
```

I/O (input/output)

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

int main()
{
    cout << "Please give me a name " << endl;
    string name;
    cin >> name;
    cout << "Hello " << name << "!" << endl;
}
```

Input

```
type someVariable;  
cin >> someVariable;
```

Skips leading whitespace.

Reads into **someVariable** until whitespace or type mismatch.

```
string someString;  
getline(cin, someString);
```

Does not skip leading whitespace.

Reads into **someString** until end of line.

Next reading starts on the following line.

Percentages

Ask the user for a numerator, ask the user for a denominator and print the fraction as a percent.

Numerator: 1
Denominator: 5
20%



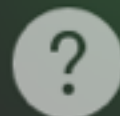
9:41

“What's $0 \div 0$ ”

tap to edit

Imagine that you have 0 cookies and you split them evenly among 0 friends. How many cookies does each person get? See, it doesn't make sense. And Cookie Monster is sad that there are no cookies. And you are sad that you have no friends.

$0 \div 0 =$
indeterminate



“What is $0 \div 0$ ”

tap to edit

$0 \div 0 = \text{indeterminate}$



Cookie Monster @MeCookieMonster · Jun 30

Me no know how to divide by zero, but me know how to divide cookies into crumbs!



1.3K



1.2K







WYNN TANNER
Director of Marketing

Campaign

Updates (9)

Comments (366)

★

Share this project

September 15

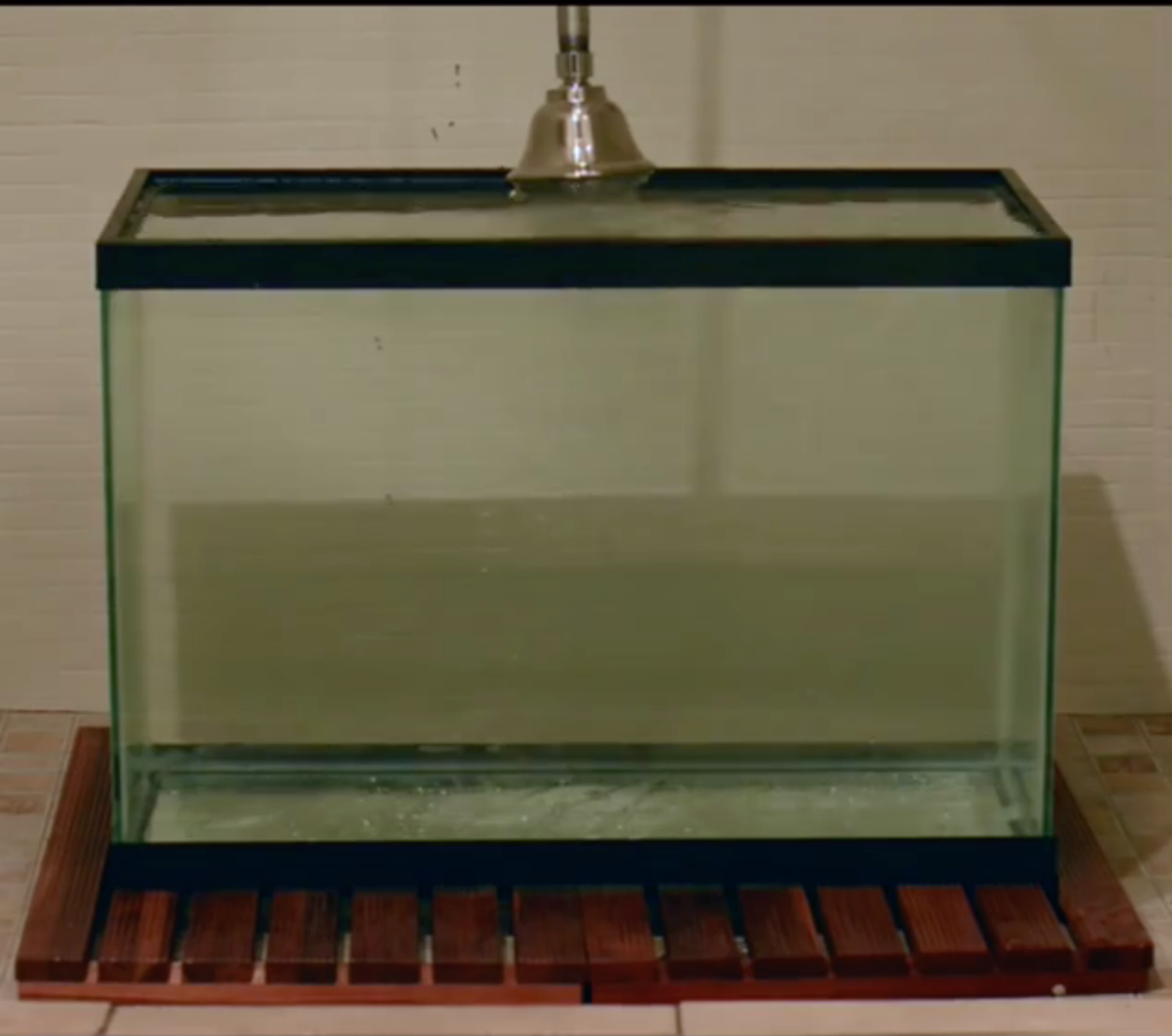
Thank you. Gracias. Xiexie.
Danke. Merci. Arigato.

What an incredible month it has been! With your help we received over 3000% of our funding goal, making Nebia one of the top 25 most funded projects on Kickstarter... [Read more](#)

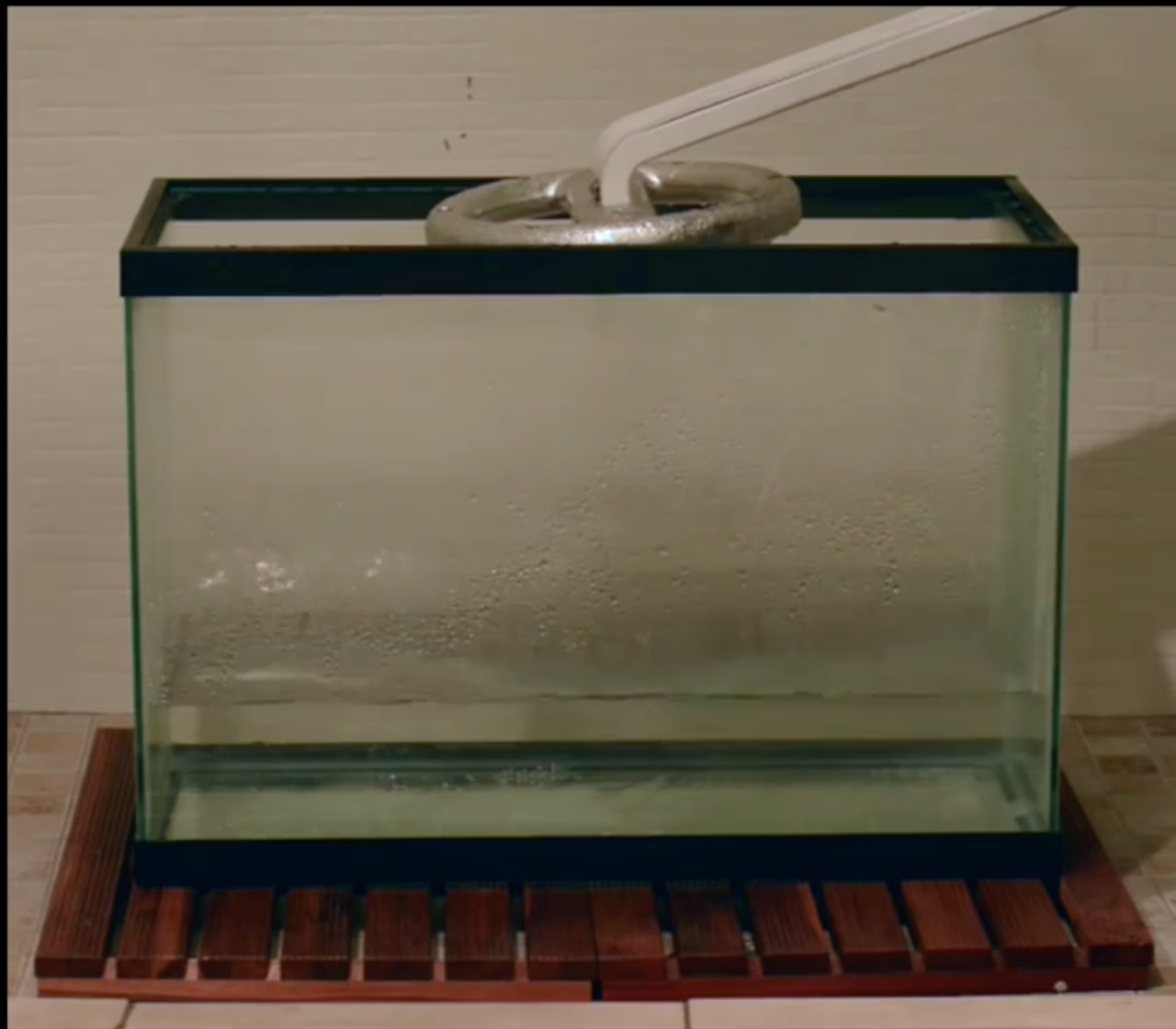
5 Comments 31 likes

September 11, 2015

Successfully raised \$3,126,115 USD with 8,559 backers



Normal shower = 20 gallons



Nebia = 6 gallons

showers-0.cpp

Write a program that...

- First asks how many **minutes** the user spent in the shower.
- Then prints how many **gallons** of water were used in a “normal” shower and in a Nebia shower.
- An 8-minute shower = 20 gallons (“normal”) / 6 gallons (Nebia)

Minutes in shower: 12

You used 30 gallons of water in a normal shower.

You used 9 gallons of water in a Nebia shower.

showers-1.cpp

Now also print how many water bottles were used.

- 1 gallon = 128 ounces.
- A typical bottle of (drinking) water might be 16 ounces.

Minutes in shower: 8

You used 30 gallons of water (240 bottles of water) in a normal shower.

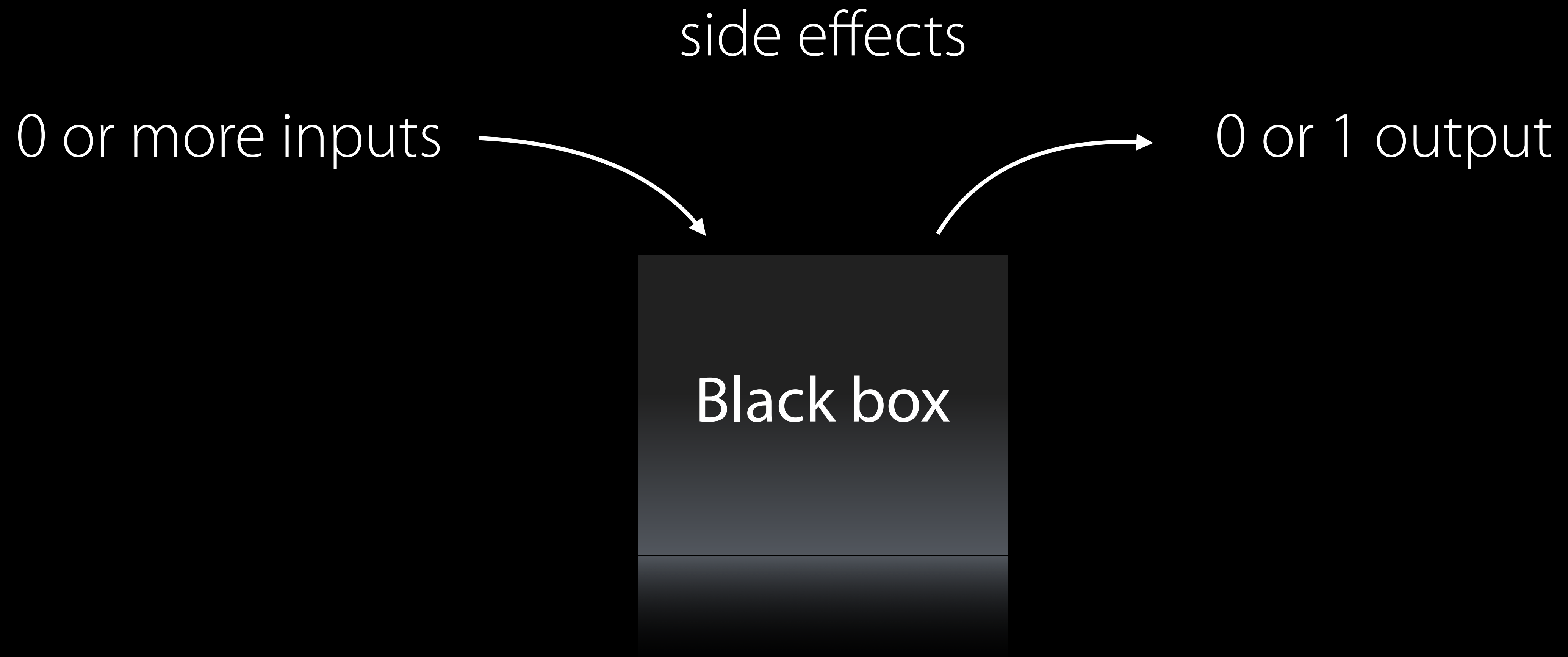
You used 9 gallons of water (72 bottles of water) in a Nebia shower.

umich.edu/~maximal/eecs183

slides, notes, source code, links, et cetera

Functions

Functions



Functions with return value

```
int bestScore = max(100, 80);
```


Functions with return value

```
int bestScore = 100;
```

Functions with return value

```
int score0 = getInt();  
int score1 = getInt();  
int bestScore = max(score0, score1);  
cout << "The best score is "  
      << bestScore << endl;
```


Functions with side effect

```
cout << "What is your name? ";  
string name;  
getline(cin, name);  
sayHello(name);
```

Why functions?

Organization

Simplification

Reusability

Nested expressions

```
double result = pow(sqrt(4), ceil(2 + 0.75));
```


Using functions

side.cpp

Deadlines

Friday

22

Project 1

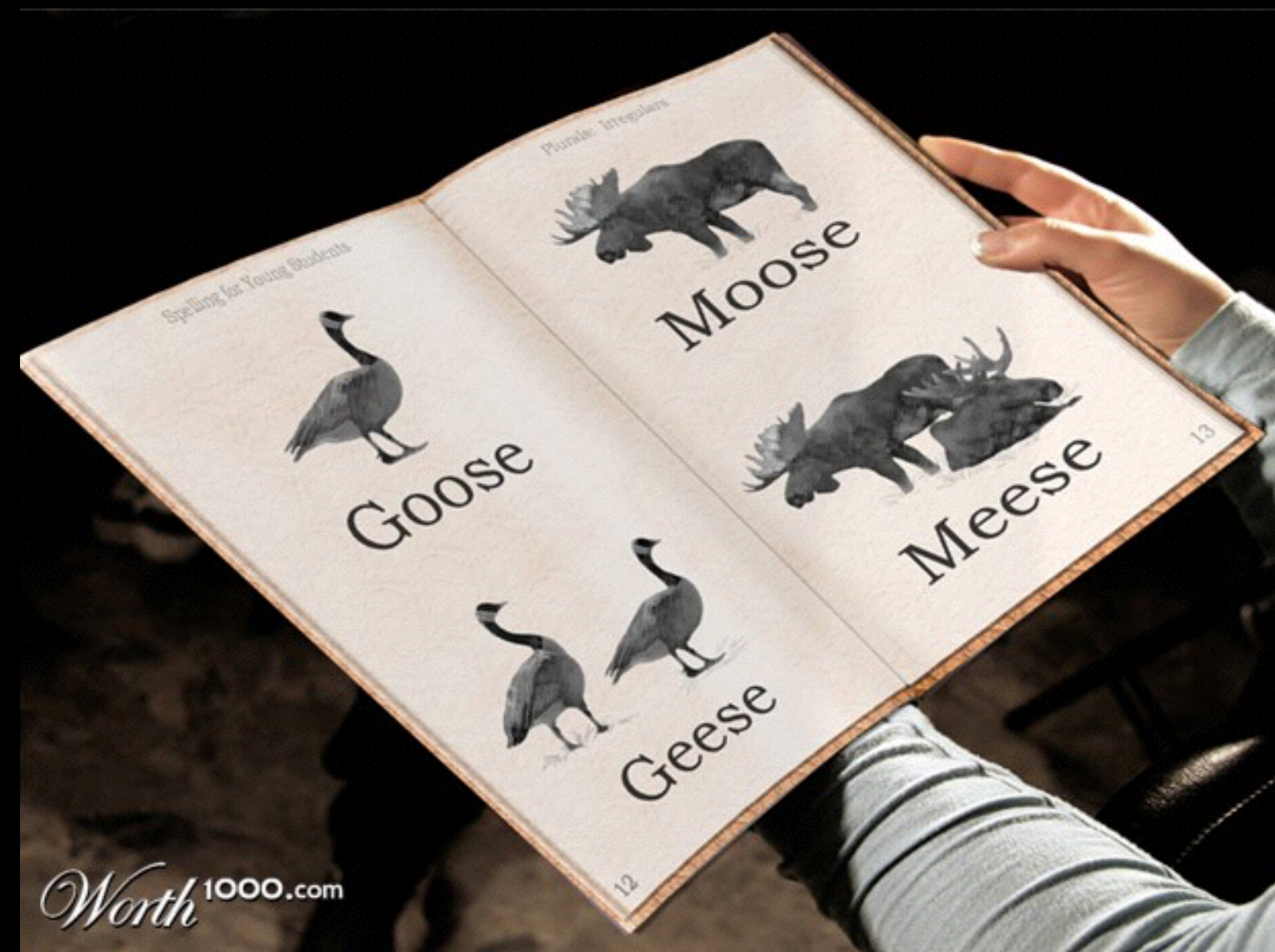
Friday

29

Assignment 2

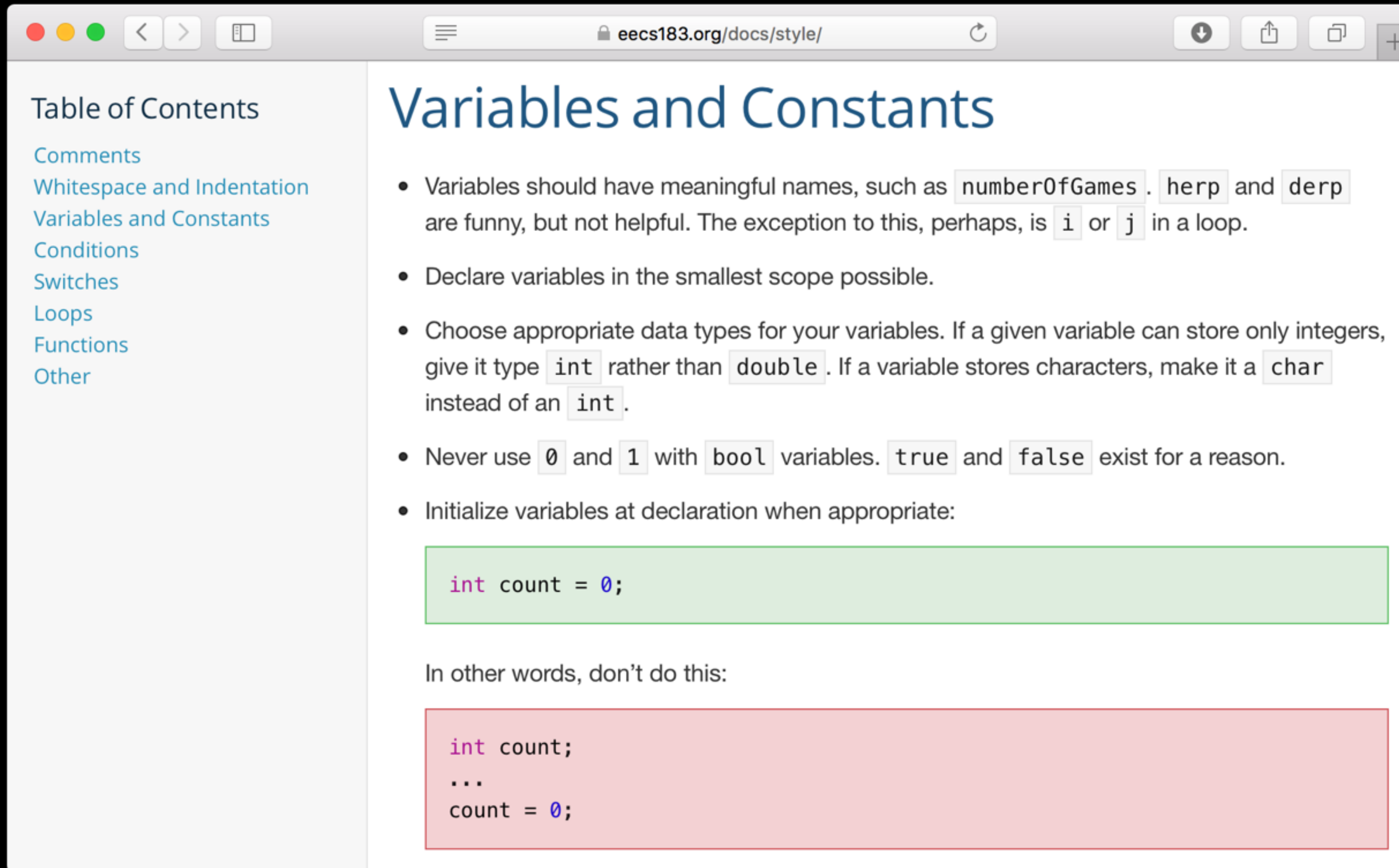
Moose or Meese?

```
cout << pluralize("moose", "meese", n);
```



Style

eecs183.org/docs/style



The screenshot shows a web browser window with the address bar displaying `eecs183.org/docs/style/`. The page has a sidebar on the left with a 'Table of Contents' section containing links to 'Comments', 'Whitespace and Indentation', 'Variables and Constants', 'Conditions', 'Switches', 'Loops', 'Functions', and 'Other'. The main content area is titled 'Variables and Constants' and contains a bulleted list of style rules. The first rule states that variables should have meaningful names, giving examples like `numberOfGames`, `herp`, and `derp`, and noting that `i` or `j` are acceptable in loops. The second rule says to declare variables in the smallest scope possible. The third rule discusses choosing appropriate data types, such as `int` over `double` for integers and `char` over `int` for characters. The fourth rule advises against using `0` and `1` for `bool` variables, mentioning `true` and `false`. The fifth rule states to initialize variables at declaration when appropriate, followed by a code example in a green box: `int count = 0;`. Below this, it says 'In other words, don't do this:' followed by a code example in a red box showing `int count;`, `...`, and `count = 0;`.

Table of Contents

- Comments
- Whitespace and Indentation
- Variables and Constants
- Conditions
- Switches
- Loops
- Functions
- Other

Variables and Constants

- Variables should have meaningful names, such as `numberOfGames`. `herp` and `derp` are funny, but not helpful. The exception to this, perhaps, is `i` or `j` in a loop.
- Declare variables in the smallest scope possible.
- Choose appropriate data types for your variables. If a given variable can store only integers, give it type `int` rather than `double`. If a variable stores characters, make it a `char` instead of an `int`.
- Never use `0` and `1` with `bool` variables. `true` and `false` exist for a reason.
- Initialize variables at declaration when appropriate:

```
int count = 0;
```

In other words, don't do this:

```
int count;  
...  
count = 0;
```

Comments

```
// calculate gallons  
int normalGallons = minutesInShower *  
                    NORMAL_GALLONS_PER_MINUTE;  
int nebiaGallons = minutesInShower *  
                    NEBIA_GALLONS_PER_MINUTE;
```

See good examples in source code posted at
umich.edu/~maximal/eecs183

Variables

```
int gallons;  
gallons = 0;
```

```
// more code...
```

```
gallons = minutesInShower * GALLONS_PER_MIN;
```

Variables

```
int gallons;  
gallons = 0;
```

```
// more code...
```

```
gallons = minutesInShower * GALLONS_PER_MIN;
```

Variables

```
int gallons;  
gallons = 0;
```

```
// more code...
```

```
gallons = minutesInShower * GALLONS_PER_MIN;
```

```
int gallons = minutesInShower * GALLONS_PER_MIN;
```


Constants

```
int normalGallons = minutesInShower * 20.0 / 8;
```

```
const double NORMAL_GALLONS_PER_MINUTE = 20.0 / 8;  
int normalGallons = minutesInShower * NORMAL_GALLONS_PER_MINUTE;
```

to be continued...