CS 103000
Prof. Madeline Blount

Week 9:
FUNCTIONS

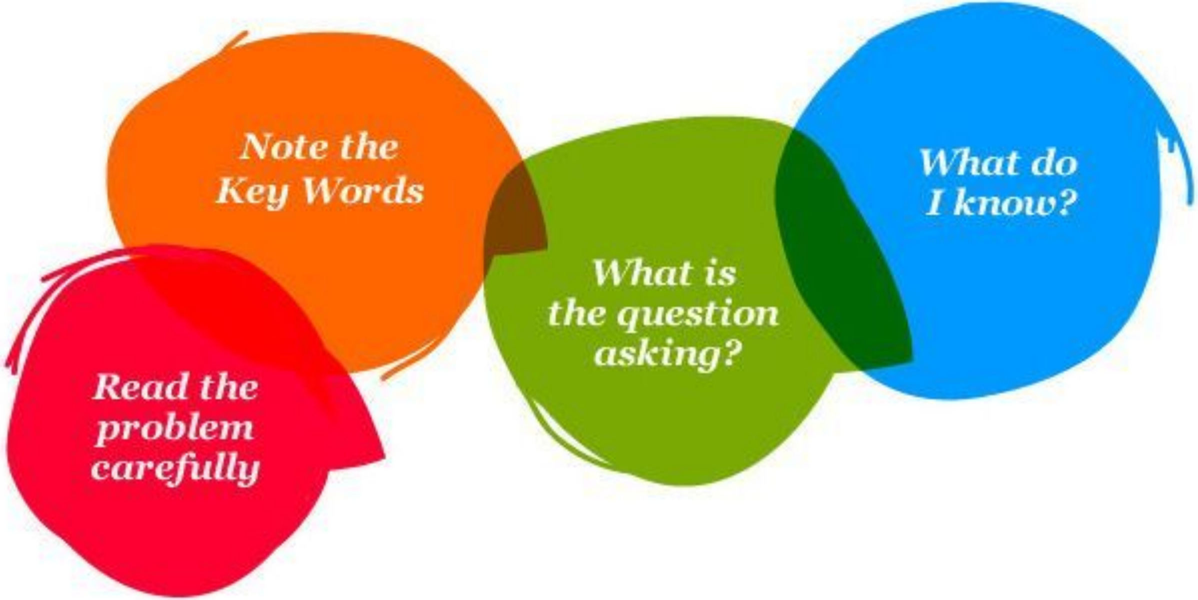
attendance link:

https://cs103-3proton.glitch.me

Dall-E 2: cats learning C++ in the forest on '90's technology

🤖 Mid-term grades + notes

- I will post adjusted grade before next class, Blackboard columns
- Also HW/reading, labs grade, and total grade so far (minus final, 20%)
- There will be extra credit opportunities! ✨

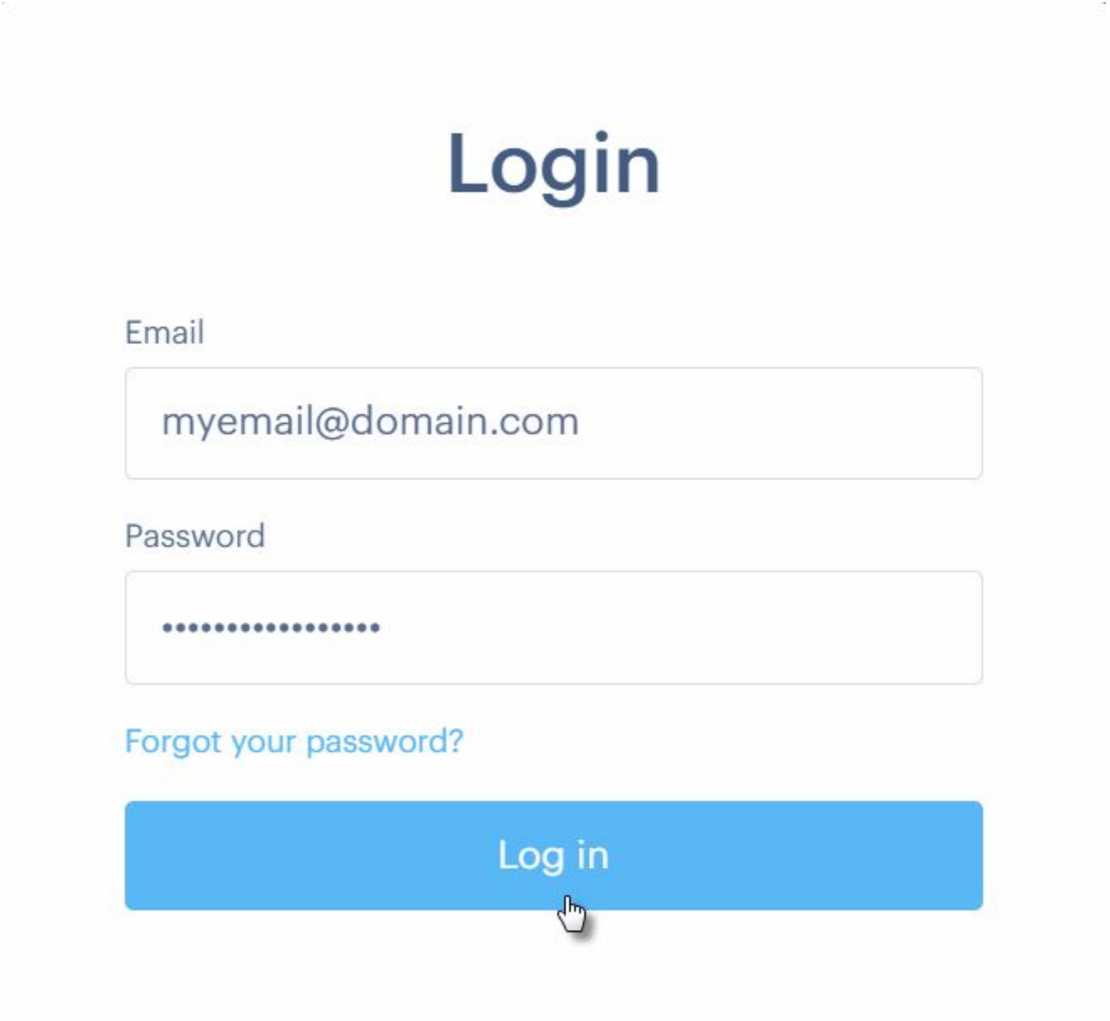


Read the problem carefully

Note the Key Words

What is the question asking?

What do I know?



Login

Email

myemail@domain.com

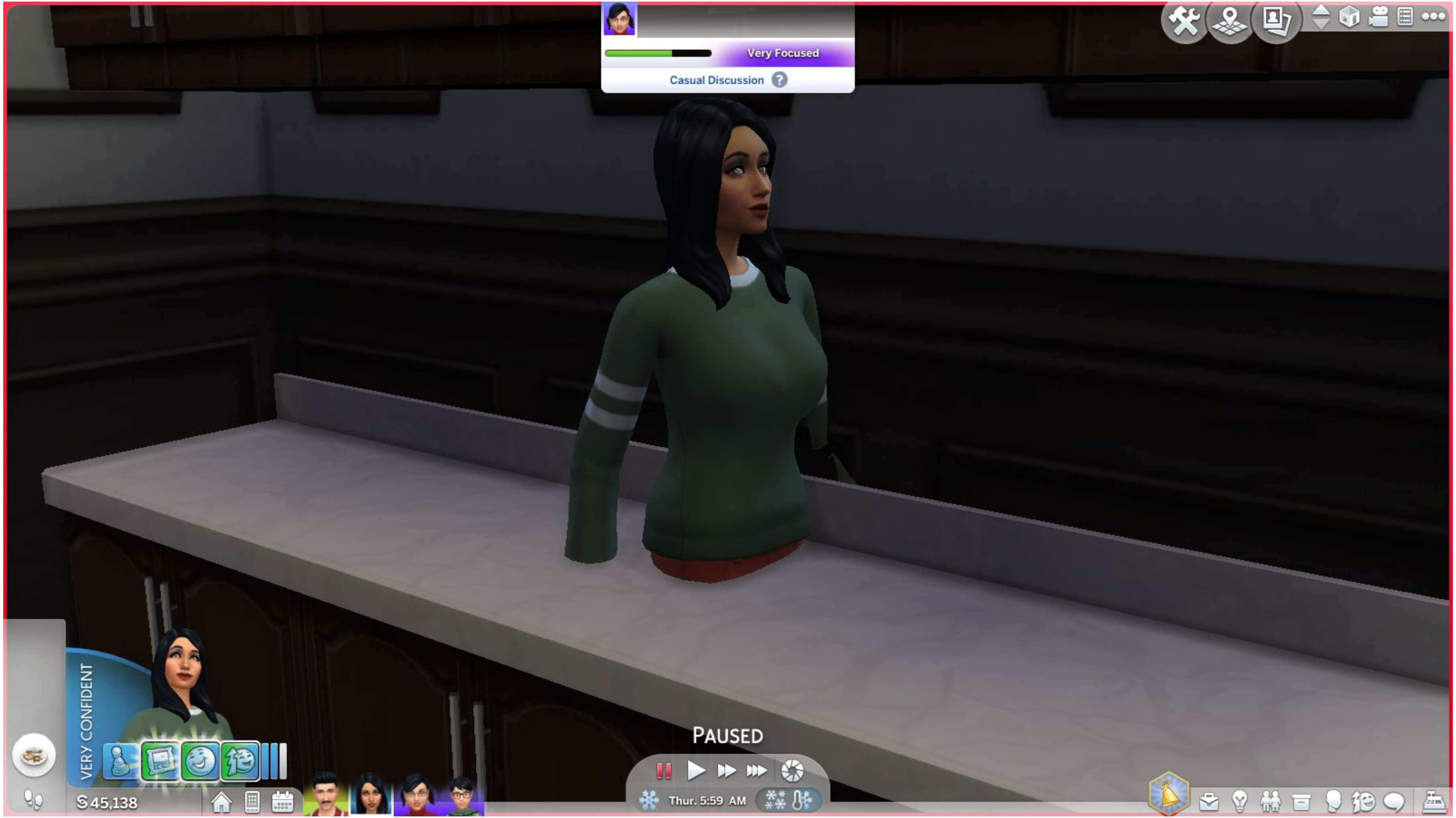
Password

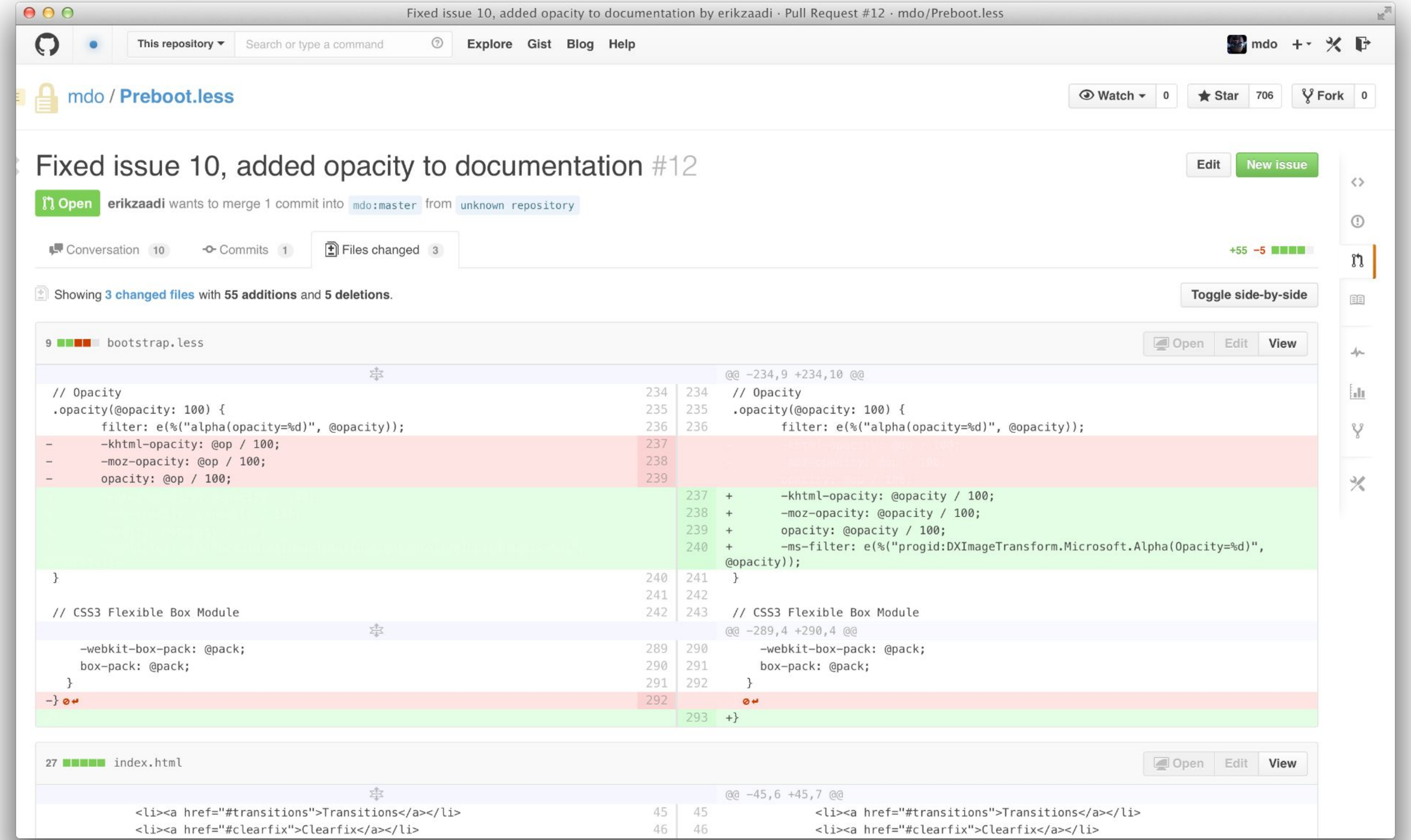
•••••••••••••••••

Vectorised Text (Post-Cleaning)

fervent

```
['estimates', 'total', 'proved', 'net', 'oil', 'natural', 'gas', 'reserves', 'filed', 'included',
'reports', 'federal', 'authority', 'including', 'natural', 'gas', 'liquids', 'natural', 'gas', 'total',
'fiscal', 'total', 'net', 'proved', 'developed', 'including', 'proved', 'developed', 'producing',
'oil', 'natural', 'gas', 'liquids', 'decreased', 'bbls', 'total', 'net', 'proved', 'developed',
'reserves', 'natural', 'gas', 'decreased', 'mcf', 'standardized', 'measure', 'discounted',
'future', 'net', 'flows', 'table', 'sets', 'estimated', 'future', 'net', 'revenues', 'total', 'proved',
'natural', 'gas', 'natural', 'gas', 'liquids', 'reserves', 'present', 'estimated', 'future', 'net',
'revenues']
```


Very Focused
Casual Discussion
VERY CONFIDENT
§45,138
PAUSED
Thur. 5:59 AM



mdo / **Preboot.less**

Fixed issue 10, added opacity to documentation #12

erikzaadi wants to merge 1 commit into `mdo:master` from `unknown repository`

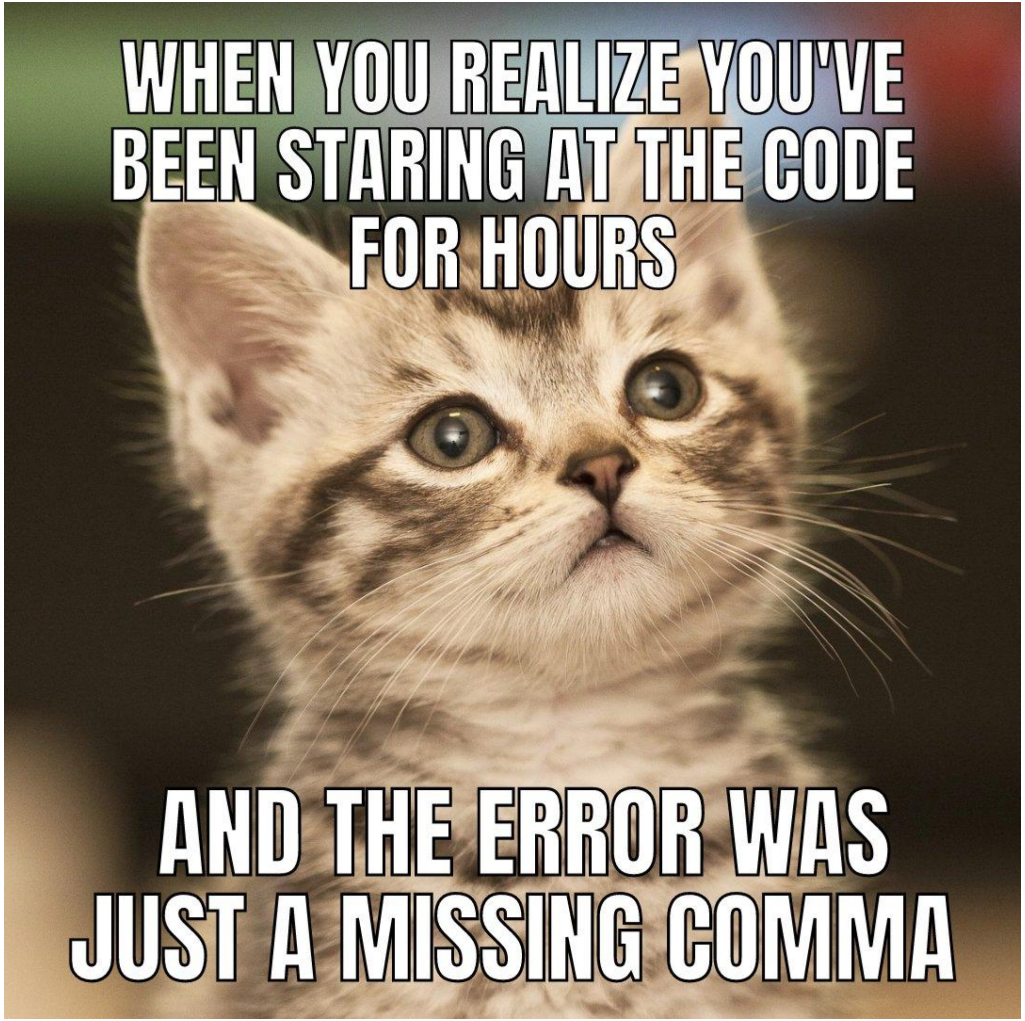
Showing **3 changed files** with **55 additions** and **5 deletions**.

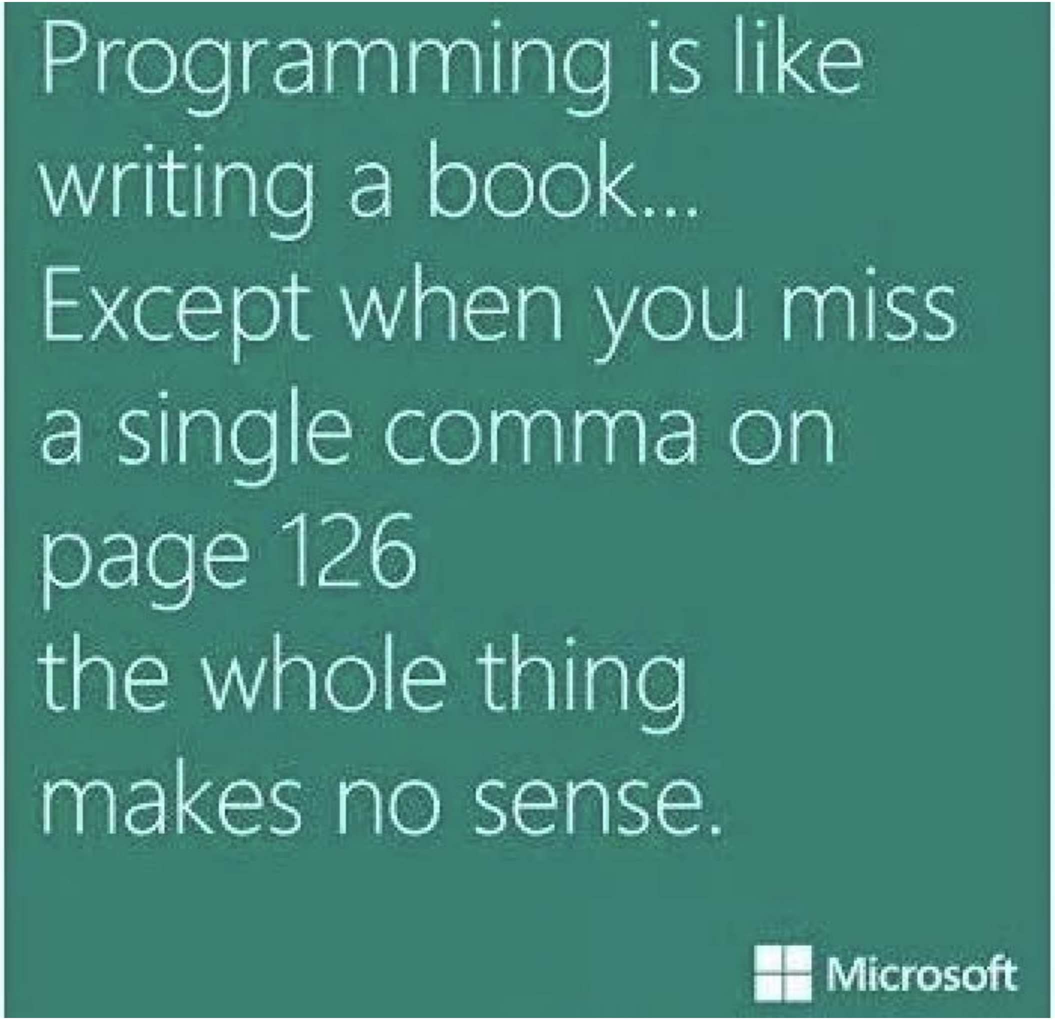
9 bootstrap.less

```
                                                                    @@ -234,9 +234,10 @@
 // Opacity                                               234   234   // Opacity
 .opacity(@opacity: 100) {                               235   235   .opacity(@opacity: 100) {
       filter: e(%("alpha(opacity=%d)", @opacity));      236   236         filter: e(%("alpha(opacity=%d)", @opacity));
-      -khtml-opacity: @op / 100;                        237
-      -moz-opacity: @op / 100;                          238
-      opacity: @op / 100;                               239
                                                              237   +       -khtml-opacity: @opacity / 100;
                                                              238   +       -moz-opacity: @opacity / 100;
                                                              239   +       opacity: @opacity / 100;
                                                              240   +       -ms-filter: e(%("progid:DXImageTransform.Microsoft.Alpha(Opacity=%d)",
                                                                    @opacity));
 }                                                       240   241   }
                                                         241   242
 // CSS3 Flexible Box Module                             242   243   // CSS3 Flexible Box Module
                                                                    @@ -289,4 +290,4 @@
    -webkit-box-pack: @pack;                             289   290      -webkit-box-pack: @pack;
    box-pack: @pack;                                     290   291      box-pack: @pack;
  }                                                      291   292    }
-}                                                       292
                                                              293   +}
```

27 index.html

```
                                                                    @@ -45,6 +45,7 @@
            <li><a href="#transitions">Transitions</a></li>    45   45             <li><a href="#transitions">Transitions</a></li>
            <li><a href="#clearfix">Clearfix</a></li>          46   46             <li><a href="#clearfix">Clearfix</a></li>
```


WHEN YOU REALIZE YOU'VE BEEN STARING AT THE CODE FOR HOURS
AND THE ERROR WAS JUST A MISSING COMMA



Programming is like writing a book...
Except when you miss a single comma on page 126
the whole thing makes no sense.

Microsoft

Programmers while reviewing the codes

my program: *works perfectly*

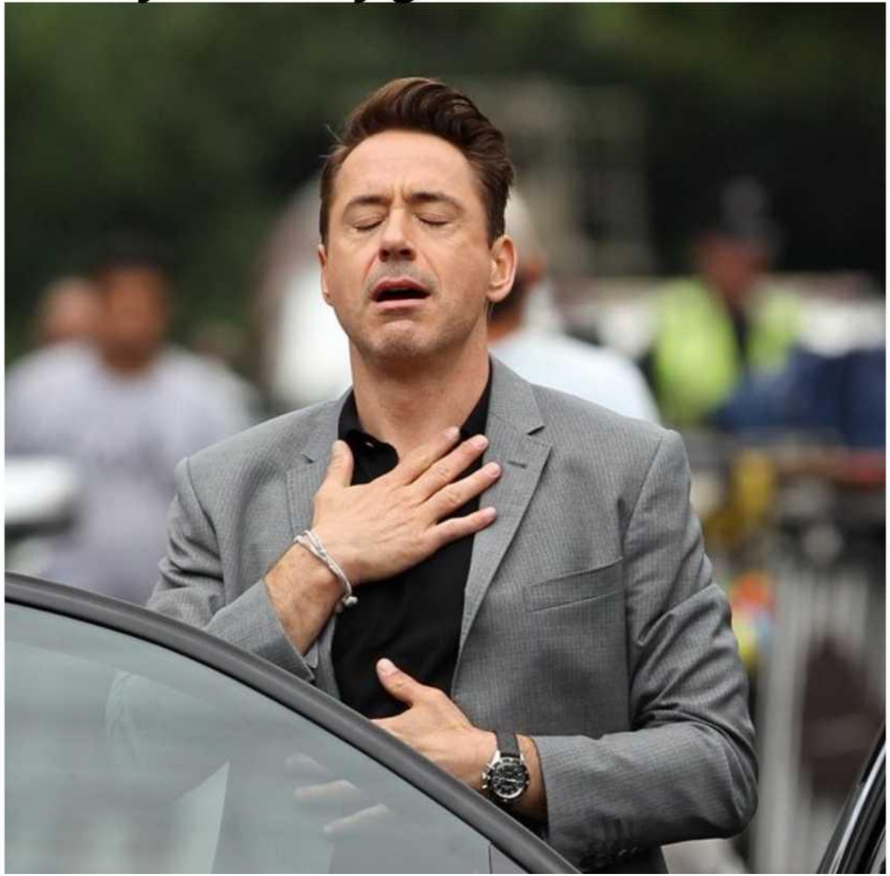
me: *cleans up the code*

also my program:

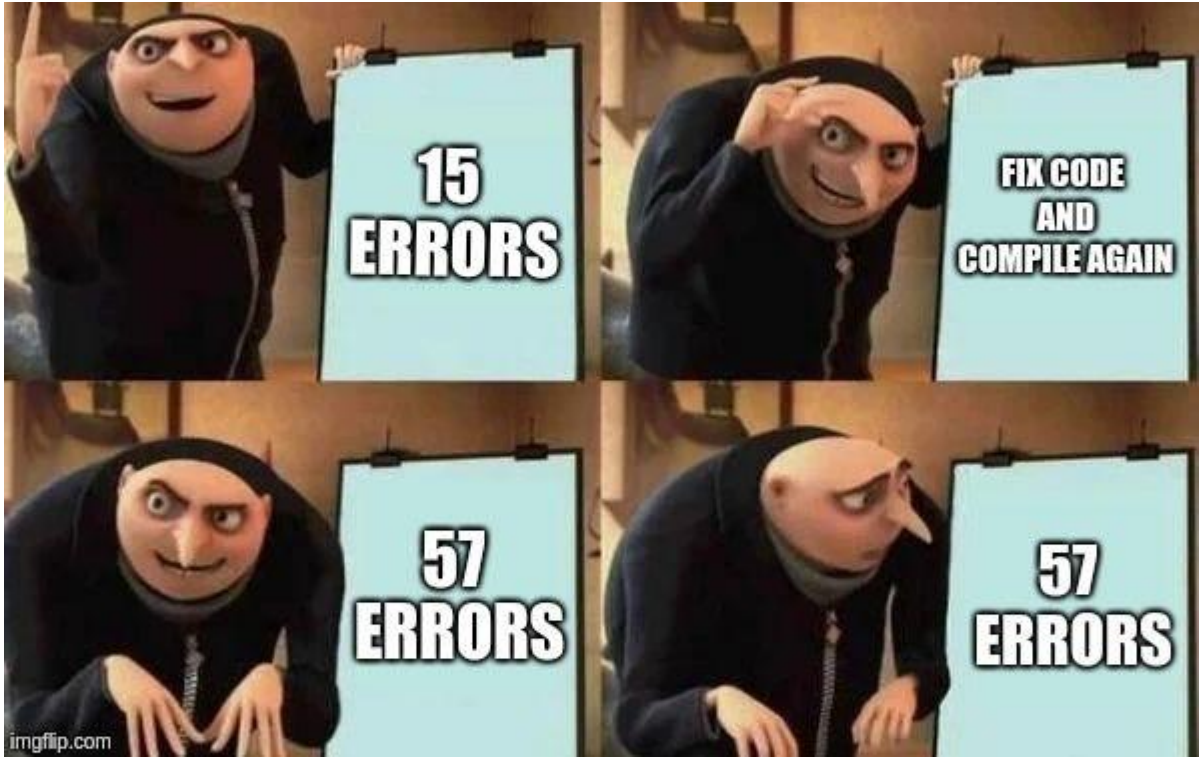
When there's a bug in your code:

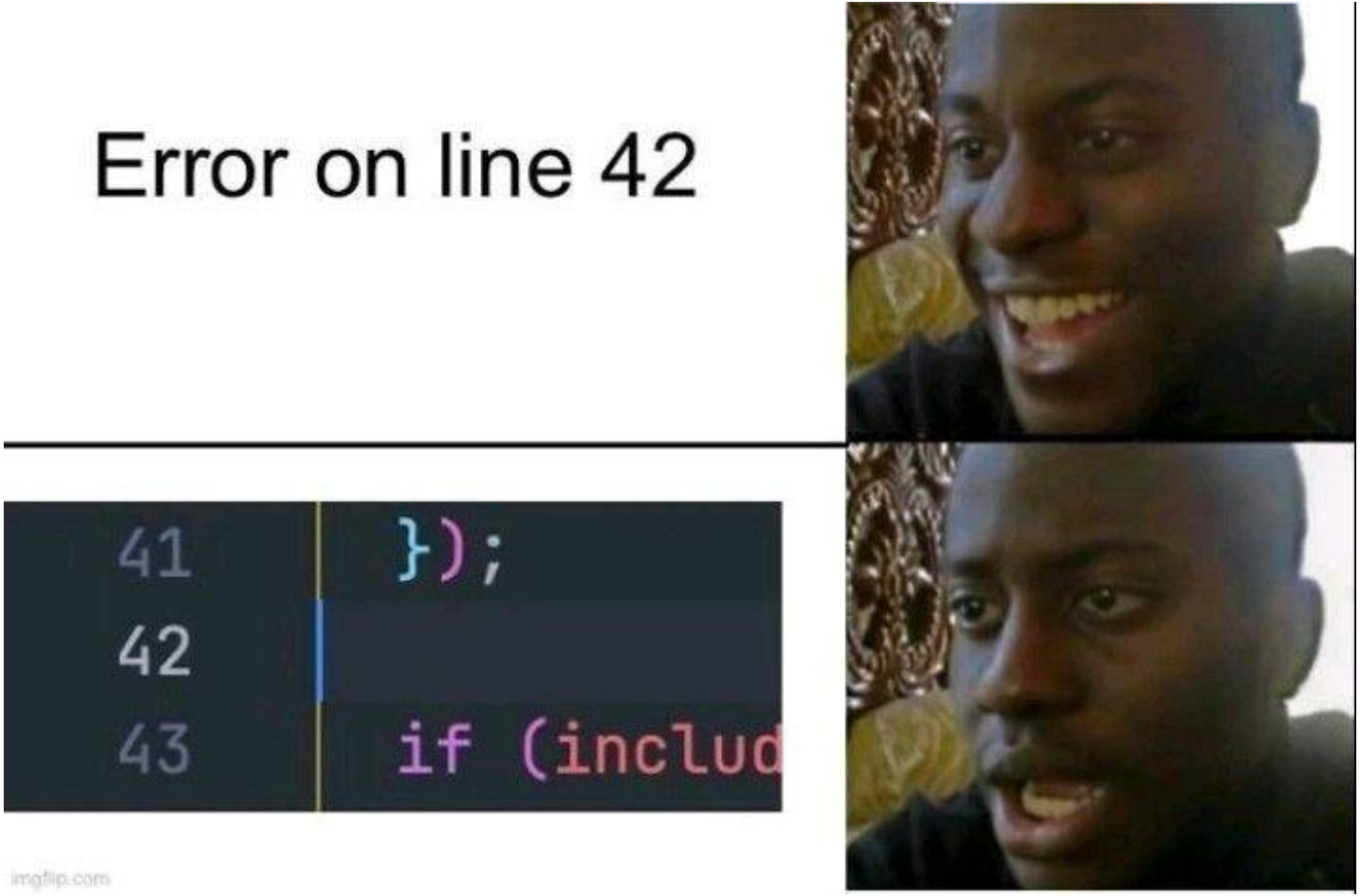
Spend 6 hours staring at your monitor with no effect
Take a walk and figure out how to fix the bug instantly
Programmers:

When you stare at your code for 2 hours and you finally get a different error



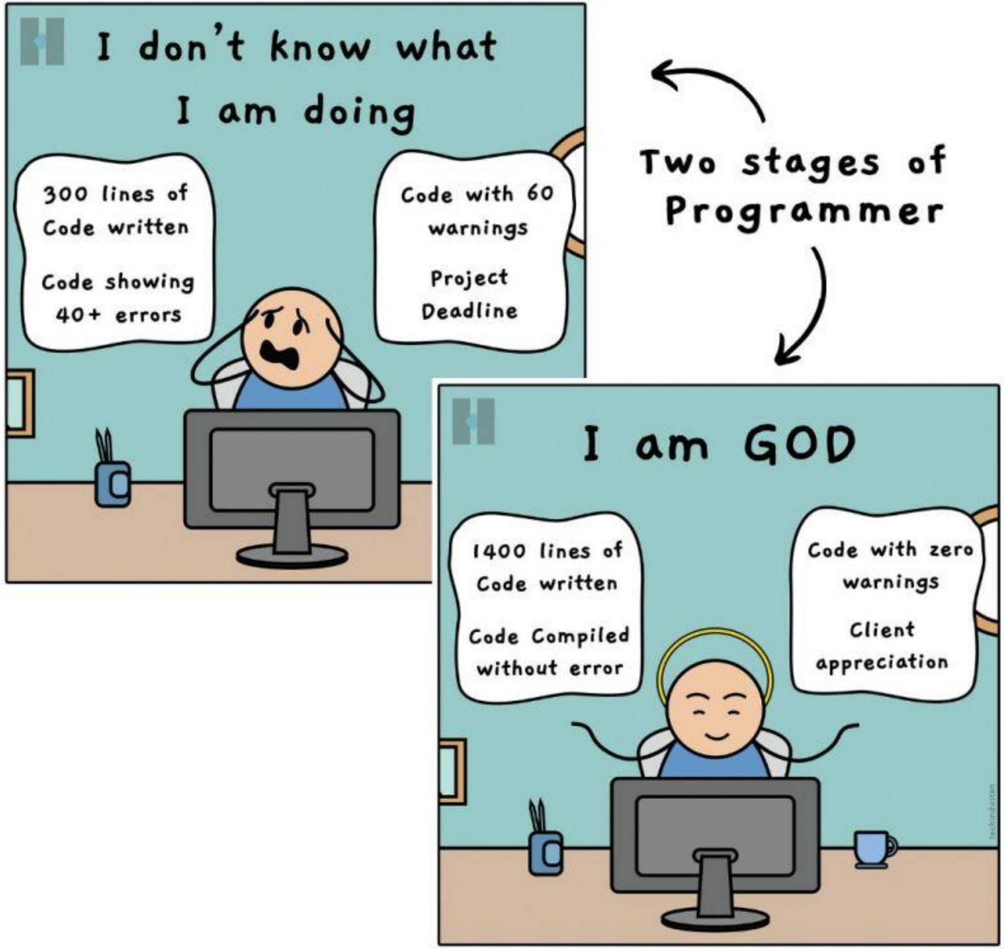
SPEND 24HRS DEBUGGING
FINALLY! A DIFFERENT ERROR MESSAGE
imgflip.com


15 ERRORS
FIX CODE AND COMPILE AGAIN
57 ERRORS
57 ERRORS
imgflip.com



Error on line 42

```
41    });
42
43    if (includ
```



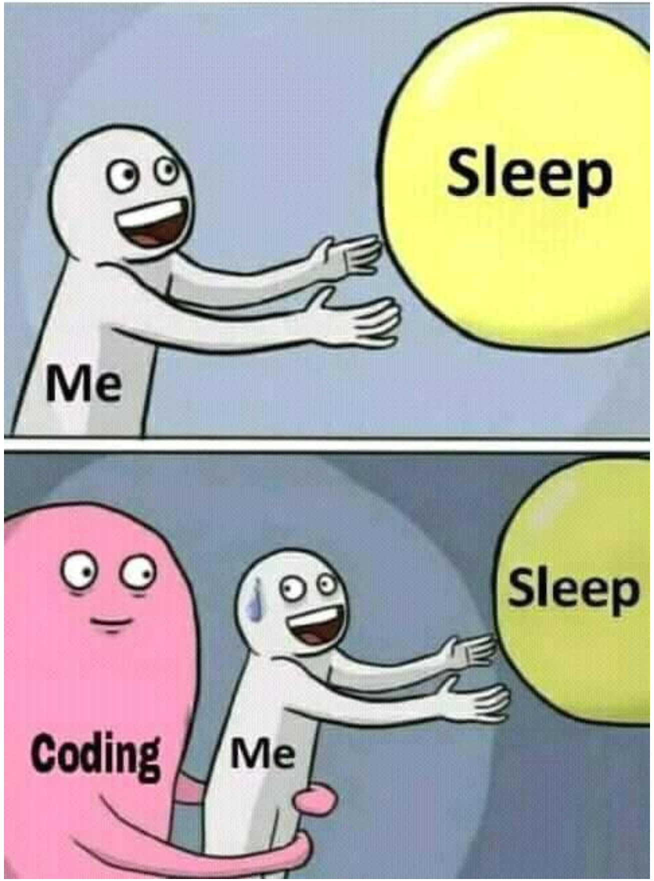
Two stages of Programmer

www.techindustan.com - Finest IT Services Company

/techindustan

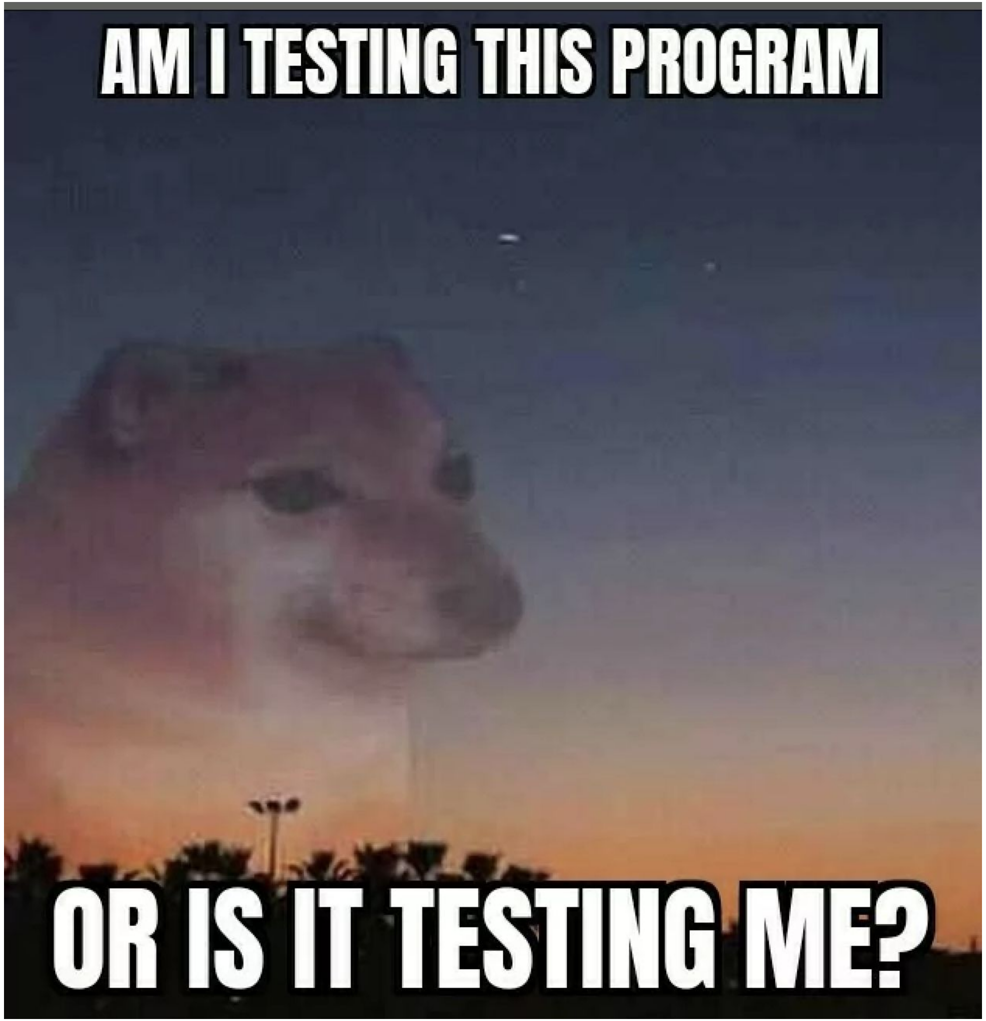
STAND BACK
WE'RE TRYING THIS IN PRODUCTION

Me every night !!



AM I BROKEN ?
I ACTUALLY ENJOY STAYING UP ALL NIGHT LONG CODING
quickmeme.com

I code better at night
11:00 PM
11:05 PM



AM I TESTING THIS PROGRAM
OR IS IT TESTING ME?

🧱 **USER-DEFINED FUNCTIONS!**

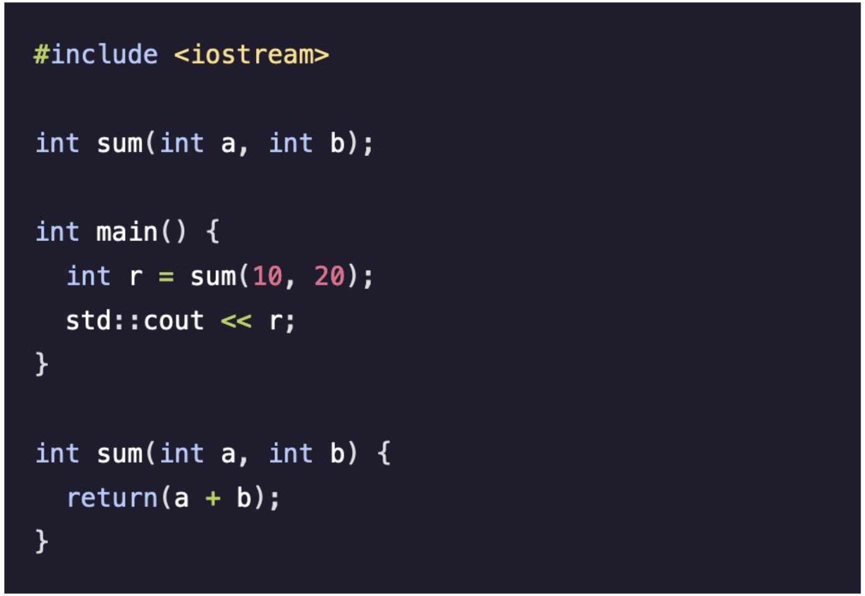
- Building blocks of code
- Set of executable statements (runs when called)

WHY?

- Easier to read (style)
- Easier to debug
- Easier to reuse, not repeat code
- Modular

Function structure:

- **1. Declare**
  (above main)
- **2. Define**
  (below main)
- **3. Call**
  (inside main)



```cpp
#include <iostream>

int sum(int a, int b);

int main() {
  int r = sum(10, 20);
  std::cout << r;
}

int sum(int a, int b) {
  return(a + b);
}
```

📣 Function declaration anatomy:

```
return type name(parameter list);
```

```
int myFunction(int var1, int var2);
```

📣 **Function definition anatomy:**

```
return type name(parameter list)
```

```
int myFunction(int var1, int var2)
{
// my code!
   return something
}
```

📣 Function call anatomy:

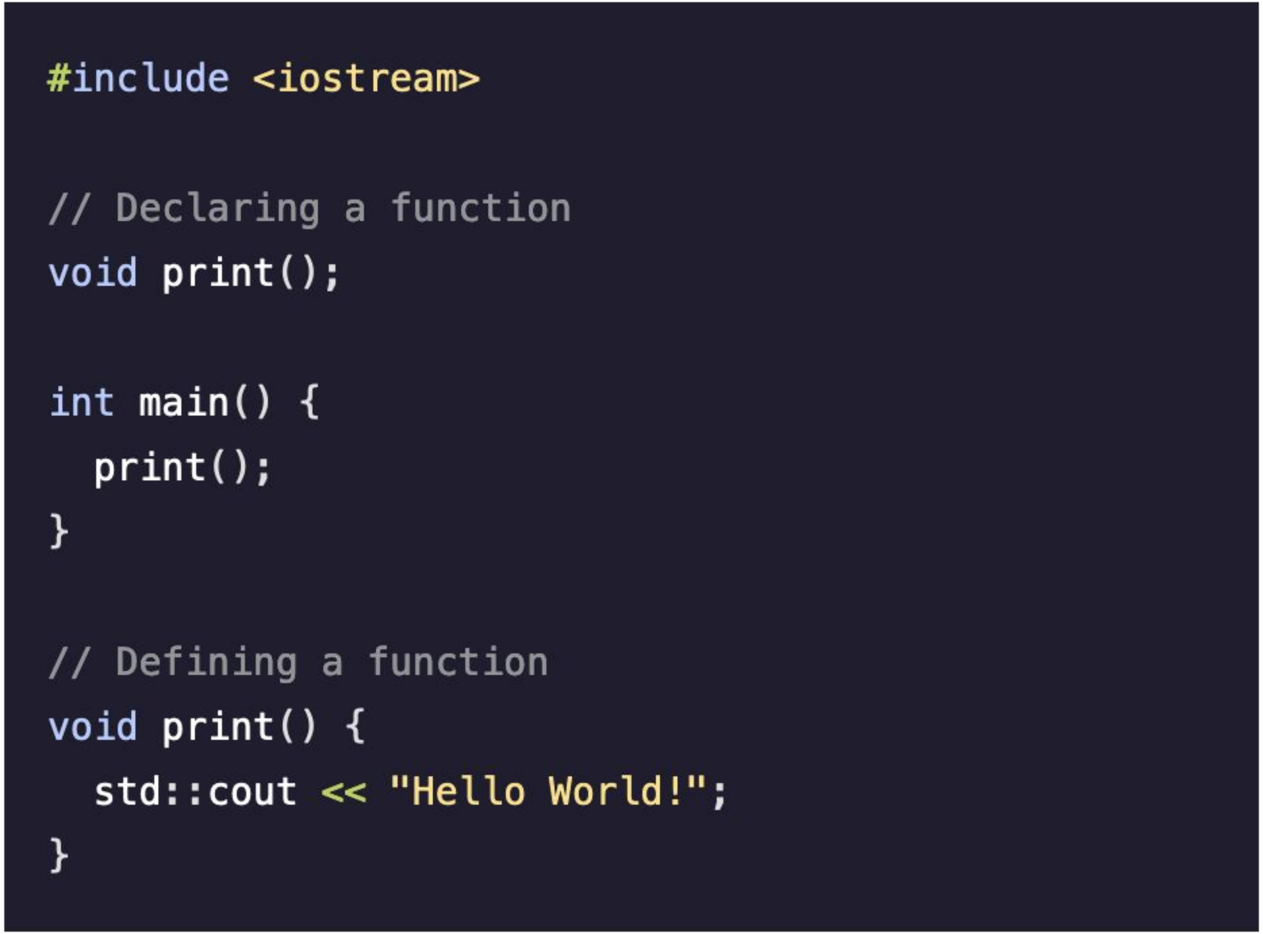
```
name(arguments);
```

```
myFunction(num1, num2);
```

📣 VOID function anatomy:

```
return type name(parameter list)
```

```
void sayHello(string name) {
// my code!
    cout << “hello, ” << name;
}
```



```cpp
#include <iostream>

// Declaring a function
void print();

int main() {
  print();
}

// Defining a function
void print() {
  std::cout << "Hello World!";
}
```

📣 MAIN function anatomy:

return type name(parameter list)

Return type = int!

0 = exit code, “success”

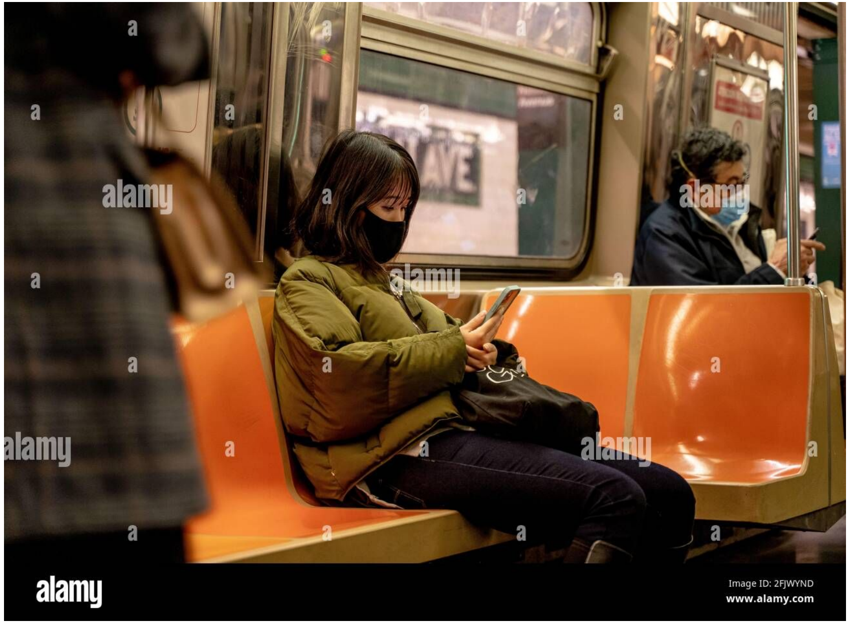
```
int main() {
// my code!
   return 0
}
```

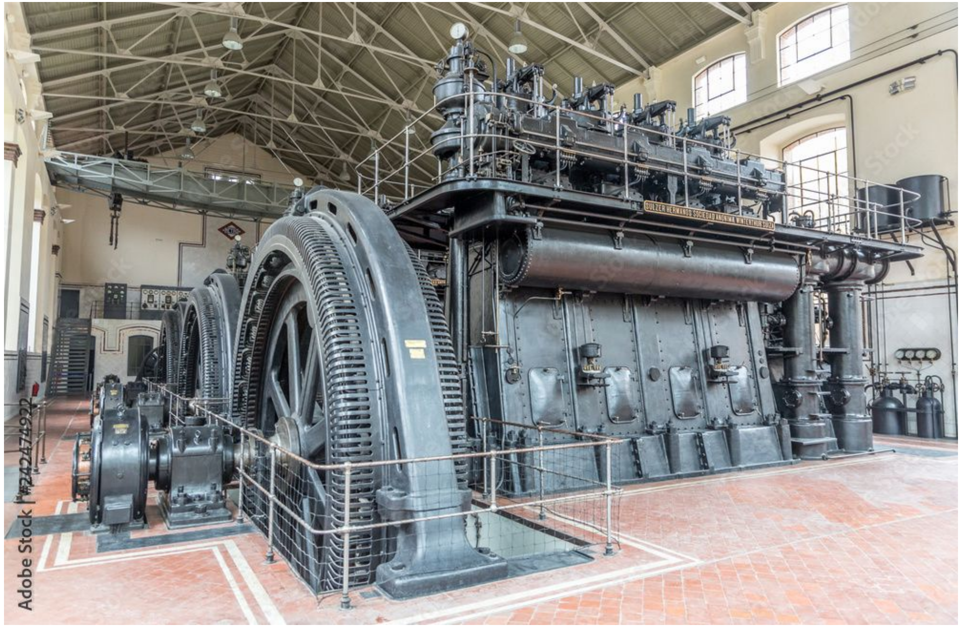
🤖 Mid-term

- **REGRET PERIOD: UNTIL FRIDAY @ MIDNIGHT**
- Speak to me after class, over Discord DM, or over e-mail; discussion + resolution

FUNCTIONS: abstraction

farther **away** from the guts, computation, data, hardware, etc.





🔭 SCOPE

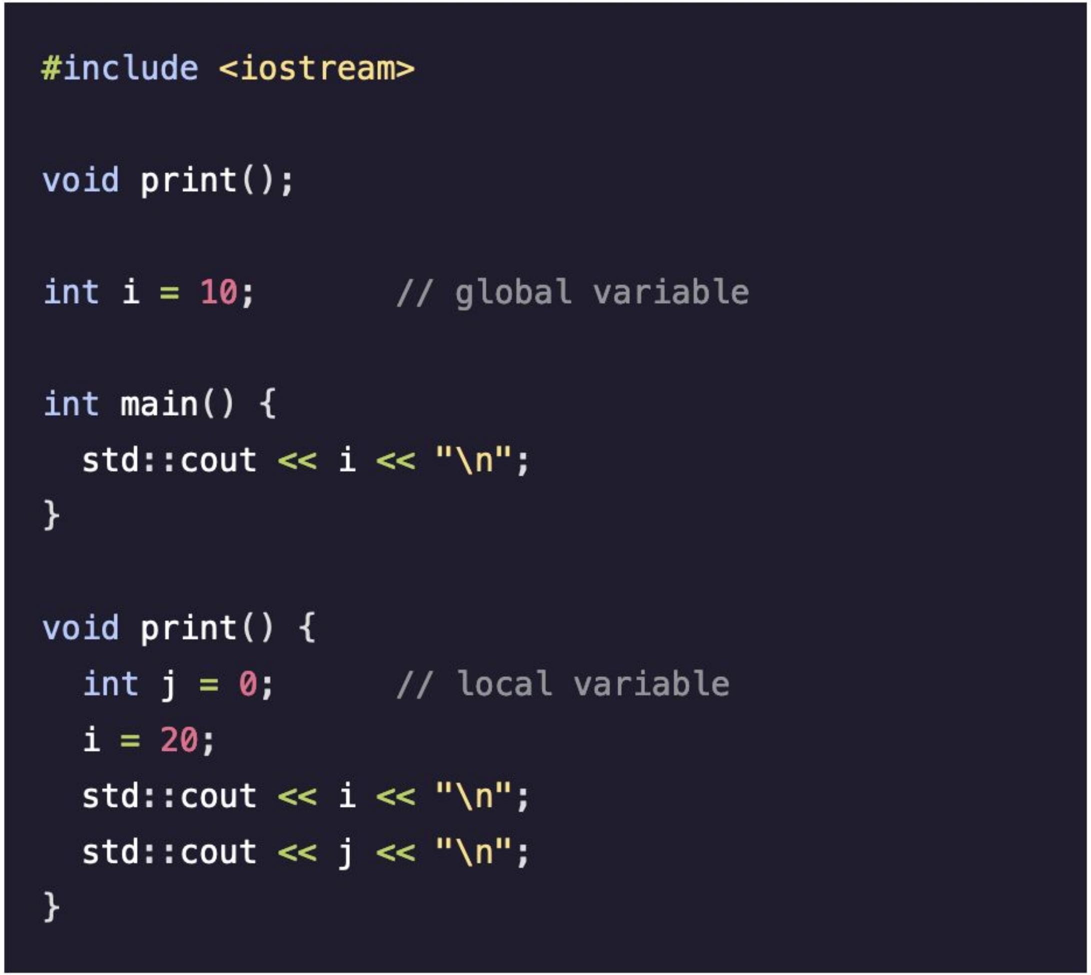
GLOBAL

everywhere!

local

only accessible in the function where they were created

🌍



```cpp
#include <iostream>

void print();

int i = 10;        // global variable

int main() {
  std::cout << i << "\n";
}

void print() {
  int j = 0;      // local variable
  i = 20;
  std::cout << i << "\n";
  std::cout << j << "\n";
}
```

“Function overload”:

Functions can have the same name but handle different data types

```
int cubeNumber(int x);
double cubeNumber(double x);
```

"<u>PASS BY VALUE</u>":
everything we have done so far, local variables stay local, because every parameter is a "copy"

"<u>PASS BY REFERENCE</u>":
can treat parameter like a global variable, changes outside of scope

pass by reference
cup =
fillCup( )
pass by value
cup =
fillCup( )
www.penjee.com