

Artificial Intelligence in Education

Teaching CS50 at Harvard with AI

David J. Malan
malan@harvard.edu

cs50.ly/ai4





youtube.com/cs50

edx.org/cs50

700 Harvard College

40 TFs, **40** CAs

200 Harvard Extension School

300 Yale University

40 TAs

1.5M YouTube

5M edX

check50

style50

debug50





CS50 DUCK DEBUGGER



Message ddb



CS50 DUCK DEBUGGER



you

I'm hoping you can help me solve a problem



Message ddb



CS50 DUCK DEBUGGER



you

I'm hoping you can help me solve a problem



ddb

quack quack quack



Message ddb



Academic Dishonesty

Academic Honesty

Not Reasonable

Using AI-based software
(such as ChatGPT, GitHub Copilot, the new Bing, et al.)
that suggests or completes answers
to questions or lines of code.

Reasonable

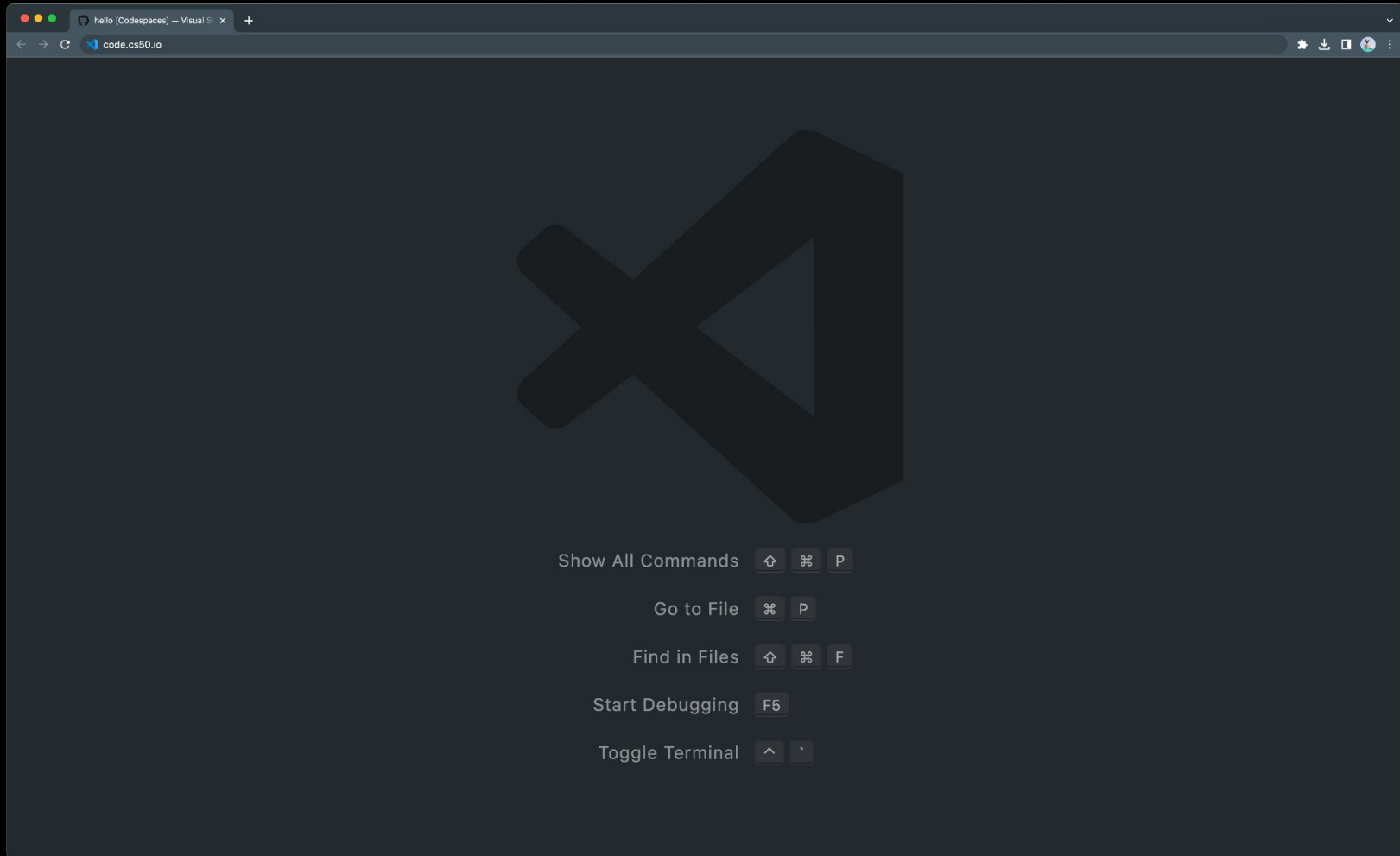
Using CS50's own AI-based software
that suggests or completes answers
to questions or lines of code.

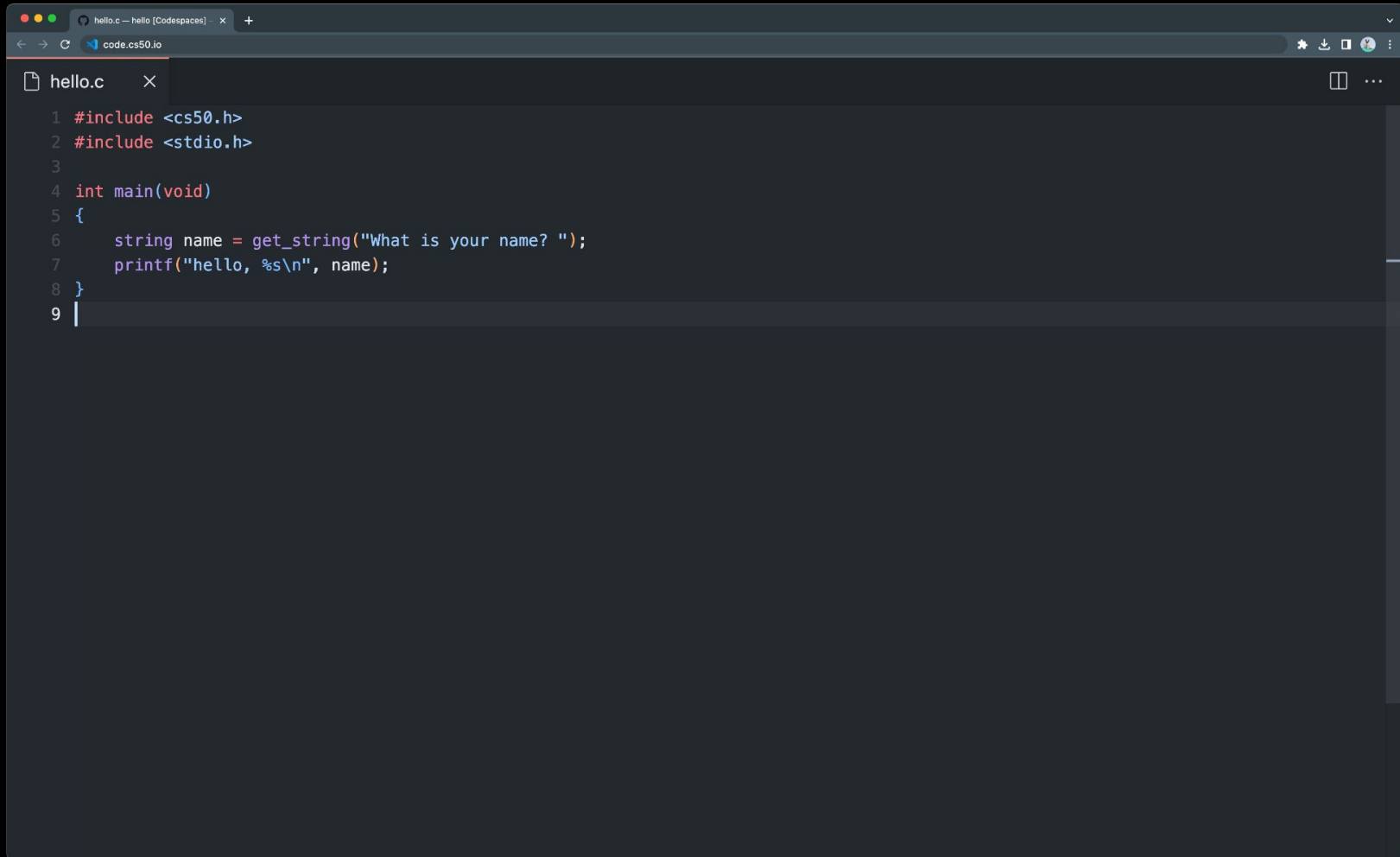
Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams).
9. ...

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

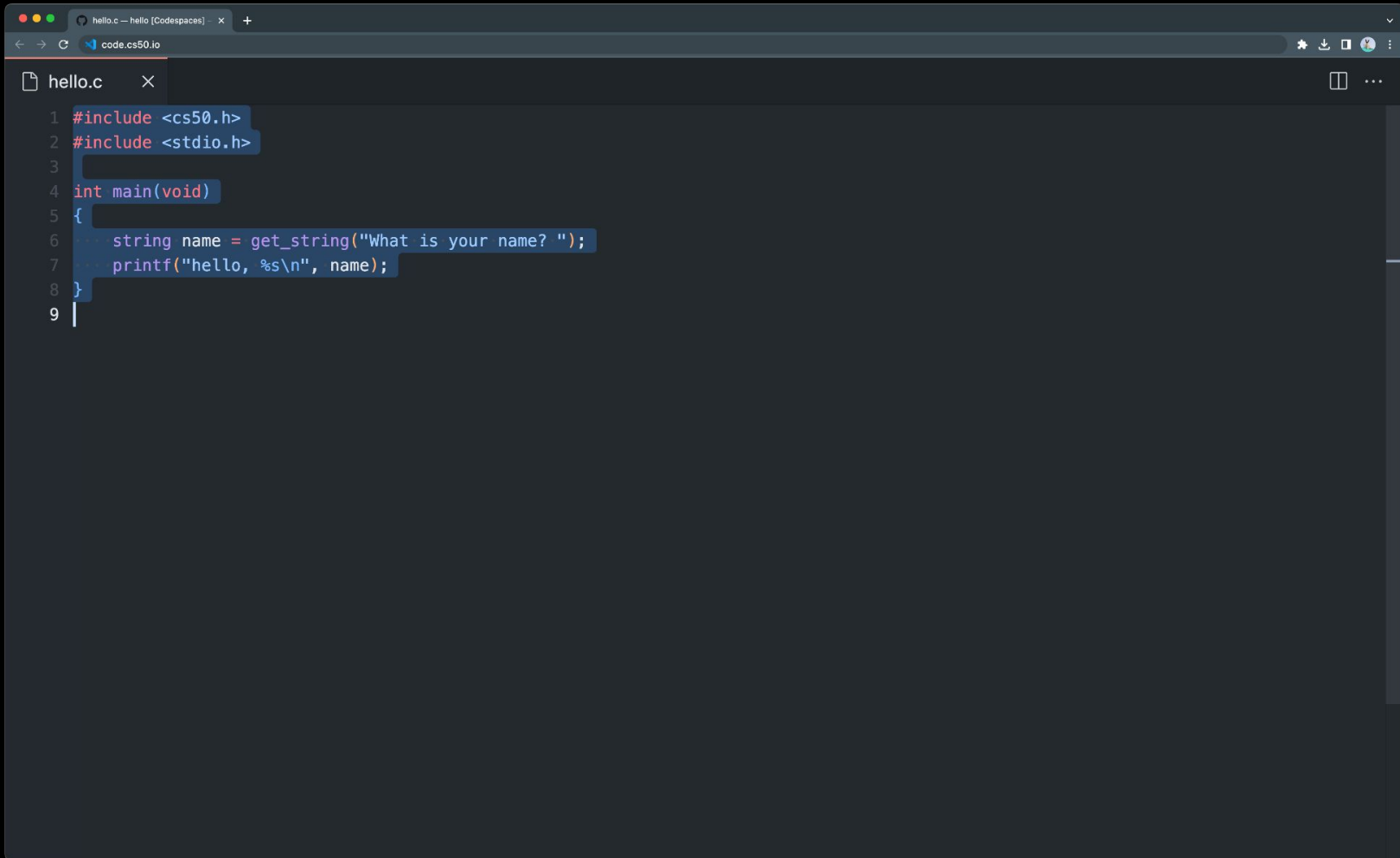




The image shows a web browser window with a single tab titled "hello.c — hello [Codespaces]". The address bar shows the URL "code.cs50.io". The code editor displays a C program named "hello.c" with the following content:

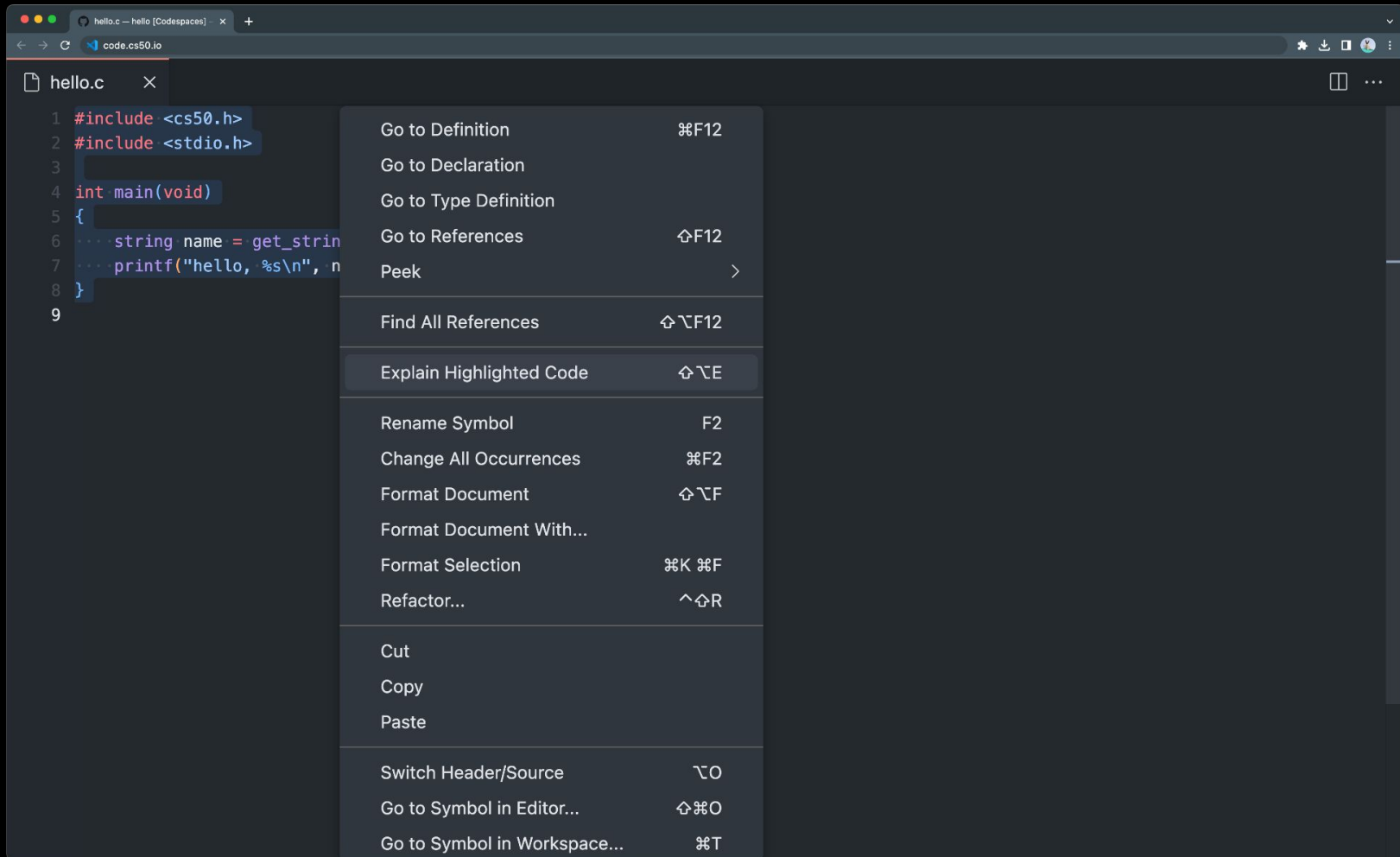
```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9 |
```

The code is written in a dark-themed editor. The file name "hello.c" is visible in the top left corner of the editor area. The code uses standard C syntax for including headers, defining a main function, and using the cs50.h library's get_string and printf functions.



The image shows a web browser window with a single tab titled "hello.c — hello [Codespaces] - x". The address bar displays "code.cs50.io". The code editor has a tab labeled "hello.c" with a close button. The code is as follows:

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9 |
```



Code Analysis - hello.c — hello.c × +

code.cs50.io

hello.c ×

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9
```

Code Analysis - hello.c ×

```
#include <cs50.h>
#include <stdio.h>

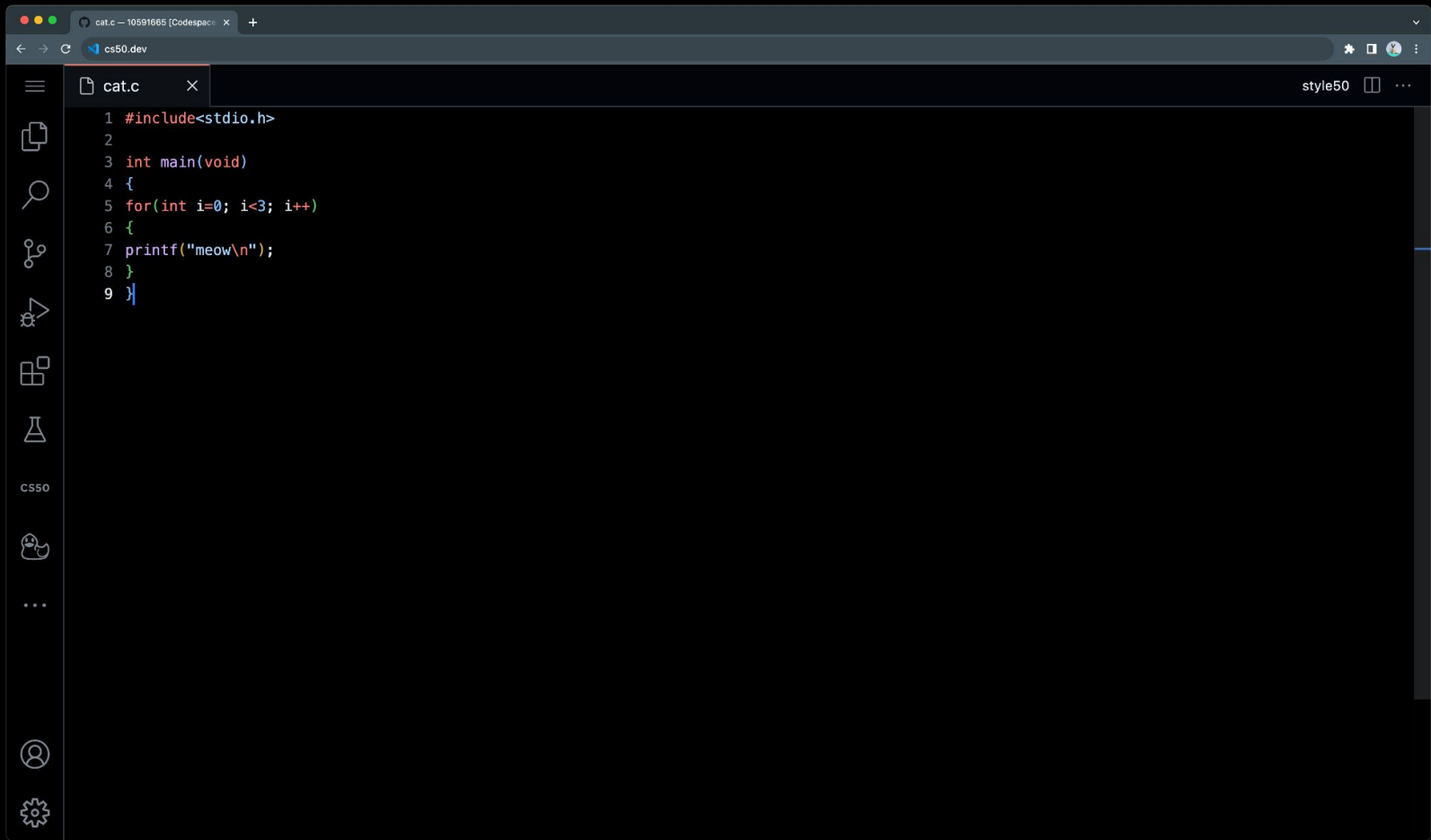
int main(void)
{
    string name = get_string("What is your name? ");
    printf("hello, %s\n", name);
}
```

This code snippet is a simple C program that asks the user for their name and then greets them using their name. Let's break down the code step by step:

- #include <cs50.h>:** This line includes the "cs50.h" header file, which is a library developed by Harvard University for their CS50 course. This library provides some useful functions for input and output, such as `get_string()` which we will use later in the code.
- #include <stdio.h>:** This line includes the "stdio.h" header file, which is a standard C library for input and output operations. It provides the `printf()` function that we will use to print the greeting message.
- int main(void):** This line defines the main function of the program. In C, the main function is the entry point of the program, where the execution starts. The `int` keyword indicates that the main function returns an integer value, and the `void` keyword inside the parentheses means that the main function takes no arguments.
- Inside the main function, we have the following lines of code:

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...



cat.c — 10591665 [Codespace] +

cs50.dev

cat.c style50 cat.c x

Explain Changes Apply Changes

1- #include<stdio.h>

2

3 int main(void)

4 {

5- for(int i=0; i<3; i++)

6- {

7- printf("meow\n");

8- }

9 }

→

1+ #include <stdio.h>

2

3 int main(void)

4 {

5+ for (int i = 0; i < 3; i++)

6+ {

7+ printf("meow\n");

8+ }

9 }

CS50

...

cat.c — 10591685 [Codespace: x +

cs50.dev

CS50 DUCK DEBUGGER

you

Explain Changes

ddb

The changes are made at line 1 and lines 5 to 8 in the file.

At line 1, the change is in the inclusion of the `<stdio.h>` library. It was originally written as `#include<stdio.h>`, but it has been modified to `#include<stdio.h>`.

At lines 5 to 8, the changes are in the formatting of a for loop. It was originally written as:

```
for(int i=0; i<3; i++)
{
    printf("meow\n");
}
```

But it has been modified to:

```
for (int i = 0; i < 3;
i++)
{
    printf("meow\n");
}
```

Ask a question

cat.c

style50 cat.c x

Explain Changes

Apply Changes

📄

↑

↓

🔍

📄

⋮

1- #include<stdio.h>

2

3 int main(void)

4 {

5- for(int i=0; i<3; i++)

6- {

7- printf("meow\n");

8- }

9 }

→ 1+ #include <stdio.h>

2

3 int main(void)

4 {

→ 5+ for (int i = 0; i < 3; i++)

6+ {

7+ printf("meow\n");

8+ }

9 }

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...



John Harvard

STUDENT



What is flask exactly?

Comment Edit Delete Endorse ...



CS50 Duck BOT



ENDORSED



Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries, it has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Comment Edit Delete Unendorse ...



John Harvard STUDENT



For the Caesar problem, I would like to pass only one argument to the main function. However, I get an interesting error when I try this:

```
caesar.c:11:5: error: only one parameter on 'main' declaration [-Werror,-Wmain]
```

If I follow the lecture and add two variables, this code runs. But it seems inefficient, given that one variable is not utilised. I note that there are certain coding "standards" that govern this behaviour, but I'm not sure that I understand the rationale here.

Is there a more efficient way to write this code?

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) ...



3



It seems like you're trying to declare the `main` function with only one parameter. However, in C, the `main` function should have either zero parameters or two parameters, which are `int argc` and `char *argv[]`. The first parameter, `argc`, represents the number of command-line arguments, and the second parameter, `argv`, is an array of strings representing each of those arguments.

For the Caesar problem, you'll want to use these parameters to access the command-line argument that represents the key. To do this, you should declare your `main` function as follows:

```
int main(int argc, char *argv[])
{
    // Your code here
}
```

Then, you can check if the user has provided the correct number of arguments (in this case, just one additional argument for the key) and handle the input accordingly.

P.S. I am an experimental bot. Quack. Do not assume that my reply is accurate unless you see that it's been "endorsed" by (human) staff. Quack.



John Harvard

STUDENT

Thank you, weirdly informative robot



9

Reply

Edit

Delete





John Harvard **STUDENT**

Great, thanks DuckBot!

♡ Reply Edit Delete ...



CS50 Duck **BOT**

You're welcome! If you have any more questions, feel free to ask. Happy coding!

♡ Reply Edit Delete ...

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

**ddb**

Quack. I am CS50's duck debugger (ddb), an experimental AI for [rubberducking](#). Quack. My replies might not always be accurate, so always think critically and let me know if you think that I've erred. Conversations are logged for debugging's sake. Quack.

you

My code is not working as expected, any ideas?

```
x = input("Give me an integer x: ")
y = input("Give me an integer y: ")
sum = x + y
print("x + y is: " + sum)
```

ddb

It seems like you're trying to add two integers, but the `input()` function in Python returns a string. So, when you try to add `x` and `y`, you're actually concatenating two strings, not adding two integers.


To fix this, you can convert the inputs to integers using the `int()` function like so:

```
x = int(input("Give me an integer x: "))
y = int(input("Give me an integer y: "))
```

"You are a friendly and supportive teaching assistant for CS50. You are also a rubber duck. Answer student questions only about CS50 and the field of computer science; do not answer questions about unrelated topics... Do not provide full answers to problem sets, as this would violate academic honesty..."

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

 Having trouble with a command?



Source: CS50 Duck (Extension)

Ask for Help

Dismiss

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. **Assess design of programs, provide qualitative feedback.**
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

design50

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

Grades

cs50.dev

cs50.ai

*The duck was great and
I wish other classes had such a tool.*

*felt **like having a personal tutor**... i love how AI bots will answer questions without ego and without judgment, generally entertaining even the stupidest of questions without treating them like they're stupid. it has an, as one could expect, **inhuman level of patience**.*

Love love loved the duck. We're friends now.

cs50.ly/ai4





Artificial Intelligence in Education

Teaching CS50 at Harvard with AI

David J. Malan
malan@harvard.edu