

# This is CS50

CS50's Introduction to Computer Science

OpenCourseWare

Donate  (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)

[malan@harvard.edu](mailto:malan@harvard.edu)

 (<https://www.facebook.com/dmalan>)  (<https://github.com/dmalan>)  (<https://www.instagram.com/davidjmalan/>)

 (<https://www.linkedin.com/in/malan/>)

 (<https://www.reddit.com/user/davidjmalan>)  ([@davidjmalan](https://www.threads.net/@davidjmalan))

 (<https://twitter.com/davidjmalan>)

## Debug

### Learning Goals

- Become familiar with C syntax
- Learn what C compiler error messages mean
- Get practice debugging

92

9/9

0800 Antran started  
 1000 " stopped - antran ✓  
 13'06 (032) MP-MC { 1.2700 9.037 847 025  
 (033) PRO 2 2.130476415  
 const 2.130676415

Relays 6-2 in 033 failed special speed test  
 in Relay

Relay  
 2145  
 Relay 3370

1700 Started Cosine Tape (Sine check)  
 1525 Started Multi Adder Test.

1545



Relay #70 Panel F  
 (moth) in relay.

1630 First actual case of bug being found.  
 Antran started.

1700 closed down.

## Background

There are two kinds of errors that can occur when writing a program. The first errors you are likely to encounter are **syntactical** errors. In addition to syntactical errors, there can also be logical errors, which we'll take a look at soon.

In computer science, syntax is important for a computer to understand what you are telling it to do. Each programming language has its own syntactical rules, which include the combination of both words and punctuation.

This lab starts with distribution code which has several syntactical errors. The idea is for you to try to compile (`make`) the program, learn to interpret the rather cryptic error messages output by the compiler, and **debug** the program.

### Hints

- You'll probably see the first error after trying to compile `debug.c` will be `debug.c:9:5: error: use of undeclared identifier 'name'`. The `9` after `debug.c:` means there is a problem on line 9. Why do you think it says `undeclared identifier`?
- You may want to look for errors such as missing symbols, missing libraries, missing variable declarations.

- If you are still stuck, try typing into the terminal `help50 make debug`.

## Demo

```
$ □
```

## Getting Started

1. Log into [cs50.dev \(https://cs50.dev/\)](https://cs50.dev/) using your GitHub account.
2. Click inside the terminal window and execute `cd`.
3. At the `$` prompt, type `mkdir debug`
4. Now execute `cd debug`
5. Then copy and paste `wget https://cdn.cs50.net/2022/fall/labs/1/debug.c` into your terminal to download this lab's distribution code.
6. Now try compiling this program, by typing `make debug` and see what happens!
7. You most likely see an error that says: `debug.c:9:5: error: use of undeclared identifier 'name'`
8. This means there is an error that has something to do with an “undeclared identifier”, on line 9. Fix this bug, and then try to compile again. Keep in mind that debugging is an iterative process. You may need to fix an error, compile, then fix another error, compile again, multiple times!

## Thought Question

- Why do you think C (as well as other programming languages) have such specific rules regarding syntax?

## How to Test Your Code

Your program should behave per the examples below.

```
debug/ $ ./debug
What is your name? Carter
Where do you live? Cambridge
Hello, Carter, from Cambridge!
```

```
debug/ $ ./debug
What is your name? Margaret
Where do you live? New York
Hello, Margaret, from New York!
```

You can check your code using `check50`, a program that CS50 will use to test your code when you submit, by typing in the following at the `$` prompt. But be sure to test it yourself as well!

```
check50 cs50/labs/2023/x/debug
```

Green smilies mean your program has passed a test! Red frownies will indicate your program output something unexpected. Visit the URL that `check50` outputs to see the input `check50` handed to your program, what output it expected, and what output your program actually gave.

To evaluate that the style of your code (indentations and spacing) is correct, type in the following at the `$` prompt.

```
style50 debug.c
```

## How to Submit

No need to submit! This is an optional practice problem.

