Errors, Exceptions, and Bugs

troubleshooting things that go wrong
when you upload and solving them

IPPS

Identify
Problem
Provide
Solution

Things that can go wrong:

it’s depressing how many ways you can find to screw things up

- (Syntax) Errors
  - You wrote ***bad code***
- Exceptions
  - Code is fine; There is a ***problem uploading***
- Bugs
  - You wrote good code that ***doesn’t work*** like you expect it to
- Wiring / Mechanical Bugs
  - duh

Syntax Errors

First - what does “Syntax” mean?

Syntax - grammar - the rules of the language.

Syntax Error:

You *broke the rules* of the language such as missing semicolons or putting values on the left side of the assignment operator.

In English it would be like saying: **I bacon? breakfast for ate** That is a nonsense sentence because the order of the words is wrong and the punctuation is both incorrect and misplaced.

Syntax Error Example

Your code doesn't **verify**. You receive the following error message:

```
Blink:33: error: expected ';' before 'delay'
   delay(1000);                          // wait for a second
   ^
exit status 1
expected ';' before 'delay'
```

Notice the parts:
Blink - the name of the code
33 - the line where the error is discovered
^ - points to the spot where the error is discovered
expected ';' before 'delay' - a description of what to fix

I ate bacon for breakfast.

Pro tips about syntax errors

1. *Start at the top* - sometimes one error can make the system think there are multiple errors. Fix the first error then verify and others might disappear.
2. *Look at the line number* in the error to find out where the error is
3. *Don't be fooled by the line number* - the number shown is when the system REALIZED the error. The cause of the error could be anywhere UP TO AND INCLUDING that line number. Frequently when there is a missing semicolon, the error will report the following line. If you forget a semicolon in line 2, the error may be reported in line 3 when another statement begins.
4. *Think broadly about what this line depends on.* If you try to declare the same variable twice, for instance, it will report the second one as problematic. However, it could be the case that the first one is unintentional. It may be anywhere.

Exceptions

These are errors that happen when you try to upload good code to the board.

If, for instance, you have the wrong serial port selected, or the board is not connected, you would get an exception. Also if you try to access a library that is in the wrong place, an exception could be thrown.

READ THE ERROR messages and look for some kind of Exception in the description. As with syntax errors, scroll to the top of the error messages to find out what is wrong.

When in doubt, search engines are your friend.

If you can't figure it out

Paste the keywords of the exception into the search box and chances are, someone else has experienced the same problem. They have probably found a solution as well.

COLLABORATE

Collaborate

Within your group or class, ask your classmates:

Have you seen this error before? What does it mean? What is the solution?

Your teacher is there to help

Ask your teacher

I did X and the result was Y ... Our group is stumped - here's what we tried and here are the searches we did so far ... can you point me/us in the right direction?

Your teacher can guide you toward the answer - if you can discover it, you'll keep the knowledge longer and be able to apply it to more similar problems.

There's an old saying:

If you give a person a fish, they eat for a day.

If you teach them to fish, they eat for a lifetime.

Bugs

Bugs

Your code verifies. No errors.

You successfully upload to the Arduino. No exceptions.

It doesn't work...

or

It doesn't work right.

the troubleshooting process

Read your code. No, really read your code.

Trace it line by line. Imagine what happens.

Keep track of values and see if what you tell it to do is what you want it to do.

the problem:
debugging is the hardest part of programming

the solution:
plan better.
write better code.

END

measure twice; cut once.