

~~CIS 162 - Computer Science 1~~

CIS 162 - Wizardry 1

Rodd Lowell & Ira Woodring



Arthur C. Clarke (a very famous science fiction writer) proposed 3 laws concerning technology:

- When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.

- The only way of discovering the limits of the possible is to venture a little way past them into the impossible.

And the one I love the most:

- Any sufficiently advanced technology is indistinguishable from magic.

I have spent a great deal of time thinking about the third law. I believe it is absolutely correct. As a matter of fact, I've noticed some things over the years that reinforce this idea and helped me develop a hypothesis.

Wizards don't do magic, they do science. They just know stuff.

The first time I realized this might be true was when I was reading *The Once and Future King* by T.H. White. It is a King Arthur story - one of the best. In it, Arthur comes across Merlyn's house in the woods. The house is separated into two parts, an upper and a lower level. The upper level is a bedroom/study. Here is the description:

<https://github.com/irawoodring/162/blob/master/introduct>



TLDR:

The contents of this room don't seem "magical".

There are books, live and dead animals, an astrolabe, even a set of encyclopedias.

Merlyn wasn't a magician, he was a scientist and a scholar!

We can see the same thing in *The Lord of the Rings*.

In the movies we get glimpses of the inside of Saruman's tower (for those of you who don't know he was a wizard gone bad!).

In it, we see exactly the same type of room Merlyn had:



Books, animals in jars, a scale for measuring.
Saruman was a scientist too.

In Harry Potter the same sort of elements appear.
The students study plants, animals, chemistry, etc.
Almost every time there is a major problem
Hermione (easily the best wizard of the three) runs
to the library to find an answer:

“Harry — I think I've just understood something! I've got to go to the library!”

And she sprinted away, up the stairs.

“What does she understand?” said Harry distractedly, still looking around, trying to tell where the voice had come from.

“Loads more than I do,” said Ron, shaking his head.

“But why’s she got to go to the library?”

“Because that’s what Hermione does,” said Ron, shrugging. “When in doubt, go to the library.”

— J.K. Rowling, Harry Potter and the Chamber of Secrets



Even in the comics this idea appears!

Jane Foster: Describe exactly what happened to you last night.

Thor: Your ancestors called it magic...

[Thor skims through a book on Norse mythology]

Thor: ...but you call it science. I come from a land where they are one and the same.

So magic is just what people attribute to phenomena they don't understand. And science helps us to understand those phenomena.

Even my own experience has reinforced this idea. I was a System's Administrator for years. It was very common for me to help someone and have them say "Wow, you're magic!"

Yes, yes I am! Because I understand the science behind it.

So yes, the name of this class - on the books - is Computer Science 1. But really, it could just as easily be called Wizardry 1.

We are going to learn the science of computers - particularly the science of communicating with them.

And thus, become wizards.

The very first, most important lesson in this field of wizardry is this:

Computers are just light switches.

They aren't magical. They aren't scary. They are machines composed of lots and lots (and lots...) of tiny switches.

We solve problems with computers by figuring out how to convert our problem into a problem a computer can solve by flipping its switches. We then interpret the state of those switches as an answer.

This is called programming.

In the early days of computing this could be tough.
Sometimes "programming" a computer meant
changing the wires around to connect components
in different ways.

These days we have higher level languages that
convert human-like ideas into commands the
computer understands for flipping switches
(transistors).

In fact, often times we convert from one language to another language, to another language... on down the line until the code is ultimately composed into instructions a computer processor can understand. We do this because each stage adds some level of abstraction that we require. This leads us to my second law of computers:

Computers are like onions. They have layers.

We need at least a cursory understanding of the layers.

Yes, I got this idea from Shrek:

VIDEO DISABLED



For instance, we might have something like this:

```
x = x + 42;
```

This seems simple to us. Take the value that x represents, add 42 to it and make x represent the new number.

A computer can't understand this though. It only knows how to flip switches and send voltage through the resulting circuit. So we need a translator program.

Our translator program may convert this into something that tells the computer to:

- flip the switches required to read the electrical voltage for the area in memory we currently have designated as "x"
- read that voltage and convert into a value that we store on a register on the processor
- store the value 42 in another register on the processor
- add the values together from the registers and store in another register
- move the value from the register back into the memory location we currently have designated as "x".

To be honest, this isn't even complete. There is a TON of work that goes on behind the scenes to make these high level human ideas like "add 42 to the value in memory location x and store back into x" into something a computer can do.

To become wizards though, we need to understand how to communicate our problems from "human-speak" to "computer-speak".

There are many, many computer languages out there. A computer language helps to translate a simple idea into something a computer can do.

In general you can do anything in one language that you can do in another. There is a law concerning this that you will learn later in your computer science career. For now just know that even though you can perform the same functions with almost any language, some are easier to solve certain types of problems.

The language we will use in this class is called Java.
Java is currently the most widely used language in
the world according to the TIOBE index:
<https://www.tiobe.com/tiobe-index/>

We will begin studying Java tonight in lab. For now, just know that it is a computer language that supports a method of communicating with computers called **Object Oriented Programming**. You will often see this abbreviated as "OO".