CS10: The Beauty and Joy of Computing

Lecture #4
Functions

Water-propelled jetpacks?! Coming this March!

You already know functions.

Functions perform an operation on a set of zero or more inputs and produce either zero or one outputs.

For a function, the same inputs MUST PROVIDE same outputs. Otherwise, we call it a procedure.

Why make functions?

They can be composed together to make even more magnificent things.

They are literally the building blocks of almost everything that we create when we program.
Why make functions?

They make it easy to repeat code.

Types of Functions

- **Command**: No outputs
- **Reporter**: Any type of output
- **Predicate**: Boolean output

Which of the following is NOT a function?

Why it Matters

- **Functions** will always produce the same output when they're given the same inputs.

- **Procedures** won't necessarily do this. They depend on things other than the function's parameters to do their job.

If a function only depends on the information it gets as input, then nothing else can affect the output. It can run on any computer or processor and get the same answer.

This makes it incredibly easy to parallelize functions. **Functional programming** is a great tool for writing software that runs on multiple systems at the same time.
Quick Preview: Recursion

Recursion is a technique for defining functions that use themselves to complete their own definition.

We will spend a lot of time on this. Fear not!