

CSCI 334:  
Principles of Programming Languages

Lecture 19: Parsing, part 2

Instructor: Dan Barowy  
**Williams**

Topics

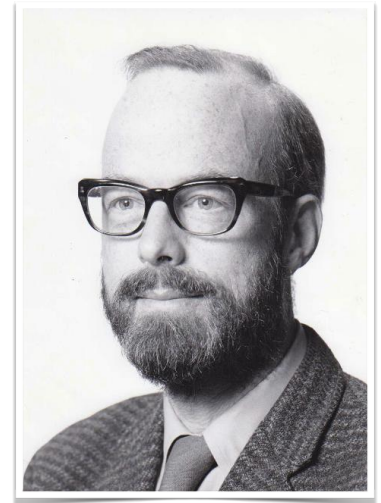
Using parser combinators

Your to-dos

1. Lab 7, **due Sunday 4/24** (partner lab)
2. Reading response, **due Wednesday 4/27**.

Hoare Property

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.” — C.A.R. Hoare



## Quiz

## More details

- It is **critical** that you read the “Parser Combinators” reading.
- I suggest that you **sit down, uninterrupted, for an hour or two**, and **work through the examples** in `fsharp`.
- The reading builds the `Parsers.fs` library that you are given for HW7.

## Example: brace language

- An *expression* is a sequence of *terms*, consisting of *at least one term*.
- A *term* is either `'aaa'`, `'bbb'`, or a *brace expression*.
- A *brace expression* is `'{'`, followed by an *expression*, followed by `'}'`.

## Example: brace language

```
<expr>  ::= <term>+  
<term>  ::= aaa  
          | bbb  
          | <brace>  
<brace> ::= { <expr> }
```

We will write a parser for this language next class.

## Recap & Next Class

### Today:

Parsing

### Next class:

Program interpretation