Subject: Grammar Specs
Posted by ffoo on Wed, 03 Feb 2016 18:23:37 GMT

I am assuming the language is going to be loosely typed? Is it correct?

Subject: Re: Grammar Specs
Posted by ffoo on Wed, 03 Feb 2016 18:40:06 GMT

Also it would be great if Dr. Lusth can list down the bare minimum entities needed for the grammar like things that can result in deduction if not covered by grammar.

Subject: Re: Grammar Specs
Posted by lusth on Wed, 03 Feb 2016 18:47:01 GMT

Look at the DPL specification.

Also, I said in class, many times, no typing (your language should be dynamically typed).

Subject: Re: Grammar Specs
Posted by ffoo on Sat, 06 Feb 2016 18:03:04 GMT

What kind of iteration we need to implement in our language? Can we just do the one based on recursion as discussed in class or we have to have loops in the language? If loops are necessary then what kind of loops are required for, while, do while etc.

Subject: Re: Grammar Specs
Posted by tscolmpton on Sat, 06 Feb 2016 19:09:59 GMT

The DPL specification says that both recursion and iteration should be implemented, which suggests that he means the imperative definition of iteration.

Subject: Re: Grammar Specs
Posted by lusth on Mon, 08 Feb 2016 15:57:33 GMT
Either a while loop or a for loop will do, as long as you can convert any loop of the other kind into your chosen loop.

Subject: Re: Grammar Specs
Posted by aeogden on Tue, 09 Feb 2016 03:42:49 GMT
View Forum Message <> Reply to Message

I was wondering if we are supposed to mirror any language we want to with our grammar or if we are supposed to make a scheme-like language. I have already started my grammar and have a lot of rules but if we are supposed to make this our own I would like to steer it more towards a java like language more than scheme but java isn't dynamically typed so I can't do that. My notes say that Dr. Lusts said we are doing a BNF grammar which is not like scheme, but scheme is dynamically typed... So I guess in class he has been mirroring C grammar? I am not familiar with C, I'm not entirely sure and slightly confused.

Subject: Re: Grammar Specs
Posted by jarobinson3 on Wed, 10 Feb 2016 20:25:15 GMT
View Forum Message <> Reply to Message

I don't understand what you are saying about BNF and Scheme. It makes no sense to me.

You can do whatever you want in your language as long as it is not trivial to implement. You can make it look like Java if you want but you just won't give types to anything.

The entire list of what you should do is here:

http://beastie.cs.ua.edu/proglan/dpl.html

Subject: Re: Grammar Specs
Posted by ffoo on Thu, 11 Feb 2016 02:10:01 GMT
View Forum Message <> Reply to Message

what should be the grammar rule for conditionList in IF or WHILE statement. Should it be expression? I am confused because expression can be primary and hence STRING or INT.

Subject: Re: Grammar Specs
Posted by lusht on Thu, 11 Feb 2016 03:54:06 GMT
View Forum Message <> Reply to Message

Make your if test an expression. You will detect non-boolean test expressions at run-time (which we will discuss in a later class).
Since our language is interpreted, could we make it so that a 'program' is just an optional statement list?

What type of comments must we implement? Should we implement both single line and multi-line comments, or is just one of those satisfactory?

This is more of a technical question: how do we represent comments in a grammar? I know there will be some terminal that will signify the beginning of a comment, but what about the contents of the comment? I was thinking it would be represented as a string, with the single line comment being terminated by the terminal NEWLINE. Is this a valid way to do it, or is there some clever way that I am missing?

Do you want to be able to run from the global scope like python or more like C where there is a single main method that gets called?

Idk what type of comments are required but comments are usually not in the grammar and are usually removed by a pre-processor.

If you make your comments C-like then you just see the first // and ignore the remaining line. Or, if you see /* you just keep going until you find the characters */ (make sure you handle escaped characters).
For simplicity, I think I want it to be more like python. That is why I was thinking of representing a program as an optional statement list. This makes a file analogous to a function, which is just an optional statement list surrounded by braces (in my language).

Subject: Re: Grammar Specs
Posted bylusth on Mon, 22 Feb 2016 01:59:51 GMT
View Forum Message <> Reply to Message

As I said in class, comments are handled by the routine that skips whitespace.