CS352

Home

Project 1: Arithmetic Expressions

Due 11:59pm Sunday Jan 16

Useful Links

- x86_64 cheat sheet (up to page 3 for this project)

Here is a small example of what we will be able to generate by the end of the project:

```
[info] Running project1.Runner 2+3
============ AST ===============
        Plus(Lit(2),Lit(3))
==================================

=========== OUTPUT =============
.text
        .global entry_point

entry_point:
        push %rbp       # save stack frame for C convention
        mov %rsp, %rbp

        # beginning generated code
```

possible. TAs or other students might help troubleshoot and solve your issue.

You can do the project on your own machine or lab machines and then upload your submission on Brightspace.

The project has been designed and tested for Linux/Mac OS. If you have only Windows installed on your laptop consider running Linux in a VM, or use the lab machines for the project. (However it should be working on Windows as well)

If you use remote access to work on your project, please use one of the lab machines data/pod1-1 to pod1-20 with the suffix cs.purdue.edu (e.g. pod1-1.cs.purdue.edu, data.cs.purdue.edu)

Download the skeleton file proj1.zip.

contain the information necessary to compile this project. You should not have to modify them.

To use sbt, launch a terminal and go to the project directory (proj1) and enter `sbt`. It will lauch an sbt console. You can run the program from there:

```
    run "arg1" "arg2" // run main program with arguments arg1 and arg2
    run "1+3*(5-8)"
    test              // run all the tests application (in src/test)
```

src/main/scala/project1/Util.scala

This file defines multiple classes that are used to generate the code and run it on your machine. Nothing needs to be modified, but it is recommanded to read it and have an idea of what is happening behind the scenes

different parsers in order to test them through the main function. You will need to modify the parser or generator you want to use.

src/main/scala/project1/Parser.scala

This class contains the definition of our intermediate language. This is the language we are using in class; please refer to the lecture for more information.

The class **SimpleParser** defined in this file is the basic parser that we are going to use for this project. It is reading a stream of characters one at a time and can extract a single digit number **getNum** or a single letter name **getName**. It does not handle whitespace.

We are keeping the parser very simple at the beginning to focus on the important concepts.

src/test/scala/project1/test/*Test.scala

These files contain some unit tests for the first parsers. You will have to write your own tests for the others. There are some functions given to you in order to make the implementation easier.

Turnin

You should turn in the **proj1** directory. Please run an 'sbt clean' and './cleanall.sh' before submitting.

To turn in your project create a ZIP file named `<purdueemailusername>-proj<N>.zip` of the `proj1` directory for example `axhebraj-proj1.zip` and upload it to the corresponding assignment on Brightspace.