

[Home / News](#)

## Lab 3: Parsing

[Schedule](#)[General Information](#)[Labs](#)

**[Update 2018-10-16 19:43]**

There exist some redundant commas and empty lines in the files under **refs-2** and **parse.c**. If you have already downloaded the previous version, please redownload the new lab environment in the below **Environment** section, and you only need to copy your **tiger.lex** and **tiger.y** to the new lab environment.

### Description

Use Yacc to implement a parser for the Tiger language. Appendix A describes, among other things, the syntax of Tiger.

You should turn in the file **tiger.y** and read the file **absyn.h**.

Supporting files available in Lab 3 include:

- **Makefile** The "Makefile" .
- **errmsg.[ch]** The ErrorMessage structure, useful for producing error messages with file names and line numbers.
- **tiger.lex** You should use \*your own\* tiger.lex in lab 2 for subsequent labs.
- **tiger.y** The skeleton of a file you must fill in.

You won't need tokens.h anymore; instead, the header file for tokens is y.tab.h, which is produced automatically by Yacc from the token specification of your grammar.

Your grammar should have as few shift-reduce conflicts as possible, and no reduce-reduce conflicts.

Furthermore, your accompanying documentation should list each shift-reduce conflict (if any) and explain why it is not harmful.

**Notice:** Before you start this lab, you should carefully read the chapter 3 of the textbook. And if you have any question about this lab, feel free to contact Dingji Li, who is the teaching assistant responsible for lab 3.

### Environment

You can download the lab environment [here](#), and then decompress it to your current directory by the following command.

```
shell% tar -zxvf lab3.tar.gz
```

Moreover, if your system lacks Lex and Yacc, you can install them with the following command (on Ubuntu).

```
shell% sudo apt-get install flex bison
```

### Grade Test

The lab environment contains a grading script named as **gradeMe.sh**, you can use it to evaluate your code, and that's how we grade your code, too. If you pass all the tests, the script will print a successful hint, otherwise, it will output some error messages. You can execute the script with the following commands.

```
shell% ./gradeMe.sh
shell% ...
shell% [^_^]: Pass #If you pass all the tests, you will see these messages.
shell% TOTAL SCORE: 100
```

## Handin

The deadline of this lab is on **Friday 12:00 AT NOON, Oct 26, 2018**, and no delay is allowed!

After you have passed the grade test, you need first package your code and rename the file lab3\_xxx.tar.gz to lab3\_[your student ID].tar.gz. For example, if your student ID is 516037900000, then the file name should be lab3\_516037900000.tar.gz, and no any other letters are included. You can use the following commands to finish this step.

```
shell% make handin #This command will do the packaging for you, generating a file named as lab3_xxx.tar.gz
shell% mv lab3_xxx.tar.gz lab3_516037900000.tar.gz
```

In the end, you need to submit the renamed tar.gz file to [ftp://dj\\_lee:public@public.sjtu.edu.cn/upload/lab3](ftp://dj_lee:public@public.sjtu.edu.cn/upload/lab3) before the deadline.

---

Go to [Top](#) // [Compilers Home Page](#)

Questions or comments regarding *Compilers* course? Send e-mail to the course Staffs or TAs.

*Last updated: Mon Sep. 17 2018*