

San José State University  
Department of Computer Engineering

# CMPE 152 Compiler Design

Section 1  
Fall 2020  
Instructor: Ron Mak

## Assignment #6

**Assigned:** Thursday, October 27  
**Due:** Monday, November 9 at 11:59 PM  
**Team assignment**, 120 points max

### Pascal Compiler

The purpose of this assignment is to give you more practice with ANTLR by completing a Pascal compiler.

Unzip file [Asgn06Cpp.zip](#) which contains a Pascal interpreter (visit methods in class `backend::interpreter::Executor`), a Pascal converter (visit methods in class `backend::converter::Converter`), and an incomplete Pascal compiler (visit methods in class `backend::compiler::Compiler` and other classes in the same namespace).

Invoke the compiler with the `-compile` command-line option. The compiler should compile a Pascal program `Foo.pas` to a Jasmin assembly program `Foo.j`.

### Complete the compiler

The incomplete compiler can compile Pascal assignment statements, **REPEAT** statements, and expressions. Complete it by adding visit methods to compile

- **WHILE** statements
- **FOR** statements
- **IF** statements
- **CASE** statements
- procedure calls
- function calls

## Test Pascal programs

Test your compiler on the following test Pascal programs to generate the equivalent Jasmin programs:

- `TestWhile.pas`
- `TestFor.pas`
- `TestIf.pas`
- `TestCase.pas`
- `TestProcedure.pas`
- `Newton3.pas`

You should get the same runtime output when you run the Pascal program with the interpreter as when you run the generated Jasmin programs after you've assembled them with the Jasmin assembler.

## What to submit to Canvas

A zip file that contains:

- All of your C++ source files and your grammar file.
- For each of the test Pascal programs, the runtime output from running the program under the interpreter, the generated Jasmin program, and the runtime output from running the Jasmin program.

Submit to **Assignment #6: Pascal Compiler**

## Rubric

Your submission will be graded according to these criteria:

Criteria	Max points
<b>TestWhile.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>
<b>TestFOR.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>
<b>TestIF.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>
<b>TestCASE.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>
<b>TestProcedure.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>
<b>Newton3.pas:</b> <ul style="list-style-type: none"><li>• Generated Jasmin program.</li><li>• Runtime output from Jasmin program.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• 10</li><li>• 10</li></ul>