

San José State University
Department of Computer Science

CS 153

Concepts of Compiler Design

Fall 2026
Instructor: Ron Mak

Assignment #4

Assigned: Thursday, September 17
Due: Thursday, September 24 at 1:00 PM
Team assignment, 150 points max

Implement Pascal type definitions

The purpose of this assignment is to give you practice implementing Pascal type definitions.

Part 1

Implement type definitions of classic Pascal's built-in **set** data type. Update the `Pc1_P2.g4` grammar file and the appropriate semantic and backend code to convert named and unnamed Pascal **set** definitions to Java's `java.util.HashSet`.

Only implement type definitions and variable declarations that involve sets. You do not need to implement statements and expressions that use sets.

Create a Pascal source file that has three proper examples of named **set** definitions in a **TYPE** section and three proper unnamed **set** definitions in a **VAR** section. Include different types of set elements. Convert the source file to an equivalent Java file.

Include at least two examples of improper **set** definitions to demonstrate error detection and recovery from syntax errors and from semantic errors (such as an improper type of elements).

Pascal reference:

http://pascal.hansotten.com/uploads/books/Pascal_User_Manual_and_Report_Fourth_Edition.pdf

Chapter 8 is about set types. Appendix D contains syntax diagrams.

Pascal set tutorial:

https://www.tutorialspoint.com/pascal/pascal_sets.htm

Part 2

Implement a new built-in **hashtable** data type for Pascal. A **hashtable** type would have a key type and an element type. The key type (used for lookups) can be any ordinal type. The element type can be any type, including another **hashtable**.

Further update the `Pc1_P2.g4` grammar file and the appropriate semantic and backend code to convert named and unnamed Pascal **hashtable** definitions to Java's `java.util.HashMap`.

Only implement type definitions and variable declarations that involve hash tables. You do not need to implement statements and expressions that use hash tables.

Create a Pascal source file that has three proper examples of named **hashtable** definitions in a **TYPE** section and three proper unnamed **hashtable** definitions in a **VAR** section. Include different types of keys and elements. Convert the source file to an equivalent Java file.

Include at least two examples of improper **hashtable** definitions to demonstrate error detection and recovery from syntax errors and from semantic errors (such as improper types of keys and elements).

Testing the type definitions

Your modified grammar and code should handle both **set** type definitions and **hashtable** type definitions. The grader will run examples of both proper and improper definitions to test your conversions to Java and to test your error recovery.

What to submit to Canvas

A zip file that contains:

- An ANTLR 4 grammar file that includes the production rules and tokens for defining **set** and **hashtable** types.
- All of your Java source files that implement defining **set** and **hashtable** types.
- Pascal source files containing proper and improper examples of defining named and unnamed **set** and **hashtable** types.
- The converted Java files.

Submit to **Assignment #4: Implement Pascal type definitions**

Rubric

Your submission will be graded according to these criteria:

Criteria	Max points
ANTLR 4 grammar file	20
Pascal examples of proper set definitions	15
Java file of converted set definitions	30
Error recovery from improper set definitions	20
Pascal examples of proper hashtable definitions	15
Java file of converted hashtable definitions	30
Error recovery from improper hashtable definitions	20