University of Malta

CSA 2010 – Compiling Techniques
Course Assignment 2005-2006

Department of Computer Science and A. I.
University of Malta.
Tutor: Kristian Guillaumier
Email: `kguil@cs.um.edu.mt`

‘Mini Script’

- The aim of this assignment is to create a simple scripting language called ‘Mini Script’.
- This assignment involves:
    - Writing BNF for the language.
    - Creating a lexical analyzer for the language (tokens include identifiers, integer constants, floating point constants, arithmetic operators like + - * and /, keyword constants like ‘Print’ and ‘For’, etc…).
    - Creating and managing a symbol table (probably a HashTable implementation).
    - Creating a top-down predictive parser for the language. The BNF for the expression language should be an augmented version of that found in the lecture slides. The parser should build an n-ary tree before it can be executed.
    - Simple error reporting.
    - The final product should be an interpreter for the language.
- Mini Scripts should be interpreted NOT compiled into executables.
- Both an interactive GUI front-end or a console-mode application are acceptable for the interpreter.
- Mini Script language features:
    - Variable declaration statements (C-like), e.g.:

      ```
      int i;

      int a, b;

      string s;

      double pi;
      ```

    - Variable declaration statements should optionally support ‘declare and assign’, e.g.:

      ```
      int i = 4;

      int b = 6, g = 10, x = -1;

      string s = “Hello World!”;
      ```

    - Integer and double constants may be negative.
    - Assignment statements with expressions on the right-hand-side, e.g.:

      ```
      i = 4 + (2 – b);

      s = “Hello” + “world!”;
      ```

  - Data types supported are **int**, **bool**, **string** and **double**.
  - Unless assigned to, variables of type *integer* and *double* have a default value of 0, variables of type *bool* have a default value of false, and variables of type *string* have a default value of “”;
  - An expression may include (and should support) identifiers. E.g. **3+counter-limit**. Whenever an identifier is encountered it should be placed in the symbol table if it is not already there (probably during scanning).
  - Variable redeclarations should not be allowed and must be reported as errors.
  - Arithmetic operators include +, -, * and /.
  - String operators include + (concatenation).
  - Operator precedence is informally defined as follows:

    Multiplication and division are higher than addition and subtraction.
  - Brackets may be used to emphasize precedence. Brackets should be supported in the language.
  - All operators are left-associative.
  - Integer constants are 32-bit integers whilst floating point constants are 32-bit floating point numbers.
  - Mini scripts should support single line comments starting with a double dash (--), e.g.:

    ```
    -- This is a comment.

    i = 4 + (2 – b);
    ```
  - Mini Scripts should support For-Loops, e.g.:

    ```
    for x = 1 to 10

    {

       println “The result is:”;

       println (i+4);

    }
    ```
  - The for loop counter variable has to declared as an integer *before* it can be used, e.g.:

    ```
    int i;

    for x = 1 to 10

    {

       ...

    }
    ```
  - The **print** and **println** statements, print an expression value to the console possibly followed by a new line (similar to *write* and *writeln* in Pascal).

- The language is case-insensitive.
- Statements end in a semicolon.
- Variable names/identifiers start with a letter or underscore symbol followed by any number of letters, underscore symbols or digits.

- The program must be accompanied by a ***GOOD*** technical report describing any implementation details and techniques used to complete the assignment. Your report must **not** include source code.
- The CD you deliver with the assignment MUST include source code and the executable version. Make absolutely sure that the executable version can run on a standard Windows XP installation (with the .NET framework and the Java Virtual Machine). If special libraries are required, include them in the installation (if legal). If you expect any problems during the deployment of the assignment, contact me.
- The CD structure should be:

  ```
  \
  \student.txt         (Text file with your full name, ID number and
                       email address)
  \src                 (Source code goes here)
  \bin                 (Executable file)
  \doc                 (Documentation in MSWORD or PDF (preferred))
  \install             (Installation IF exe in bin cannot be
                       executed without some setup (NOT preferred))
  ```

- Your assignment should include at least 4 test scripts (on the CD) showing off the functionality of the application.
- Submission via email are NOT accepted.
- The technical report *must* include the BNF for the language grammar as augmented from the lecture slides.
- ***(Important)*** Refer to the assignment instructions at **http://webster.cs.um.edu.mt/kguil/assignment.html**

An example script:

```
-- This is an example Mini Script
-- Author: Kristian Guillaumier
-- Date: 20/10/2005

-- Declare some vars.
double PI = 3.14159, area;
int radius;

for radius = 5 to 10
{
      -- Calculate the area.
      area = radius * radius * PI;

      -- Print out.
      print “The area of a circle with radius “;
      print radius;
      print “is”;
      println area;
}
```