

## CS390 Principles of Programming Language

### Assignment 3 Part I

#### Requirements:

1. Modify the Scanner to include **true** and **false** keywords, that are not id tokens, but their own token types TRUE and FALSE. An example statement using these might be:

```
isTest := true
```

2. Modify the Scanner to include int and bool keywords, that are not id tokens, but their own token types INTEGER and BOOLEAN (note keyword and type differs slightly). An example statement using these might be:

```
bool A
int B
```

3. Modify the Scanner to include a unary '!' operator with a NOT token type. For example,

```
A := !B
write !A
```

4. Modify the Scanner to handle a left (LPAREN) and right (RPAREN) parenthesis tokens. When finished it should be able to support statements like (not 'cos' is simply an id:

```
write (A * B)
C := cos(A)
```

5. Modify the Scanner to use a semi-colon to delimit a statement. For example, the following source code examples should be handled

```
read A;
read B;
sum := A + B;
write sum;
write sum / 2;
```

```
read
  beta;
sum := alpha + beta ;
```

6. Modify the Java code for the Scanner to “preprocess” Java-style comments. Tokens should not be created for such comments. Strip them out of the input code.
  - a. Single line comments that begin with “//”
  - b. Multi-line comments that begin with “/\*” and end with “\*/”.

**Submission:**

Submit your NetBeans project <yourName\_Assignment\_3\_I> to the Assignment 3 Dropbox in the Worldclass course shell associated with your current CS390 Section. (Although you will not earn points for testing, you should appropriately test your code for all requirements).

This is an **individual** programming assignment and all work, except the existing Scanner code included in the Worldclass examples, which you are expected to modify, **must** be your own.

## CS 390 Principles of Programming Languages

### *Assignment 3: Part I Rubric*

#### Assignment 3: Part I Rubric

Assignment	Exemplary	Advanced	Proficient	Not Demonstrated or Major Issues
<b>Comments</b>	Single & Multi-line comments correctly handled	Single line comment correctly handled		
<b>Keywords:</b> int, bool, true, false			Additional keywords correctly scanned	
<b>Unary Operator</b>			Additional unary Operator correctly scanned	
<b>Function</b>		Additional function keyword correctly scanned in new production		
<b>Statement Delimiter</b>			Additional statement delimiter correctly handles all statements	
<b>Deductions</b>	Submitted on time Appropriately commented Compiles correctly	Inappropriate comments 1-10%	3% deducted per day late	<b>Not</b> submitted within six days of due date or <b>does not</b> compile

© 2018 Regis University, All rights reserved

Unauthorized duplication or distribution including uploads to the Internet  
violates copyright law and various Regis University Academic Integrity policies