# Introduction to Language Theory and Compilation
# Solutions
## Session 7: Semantic Analysis

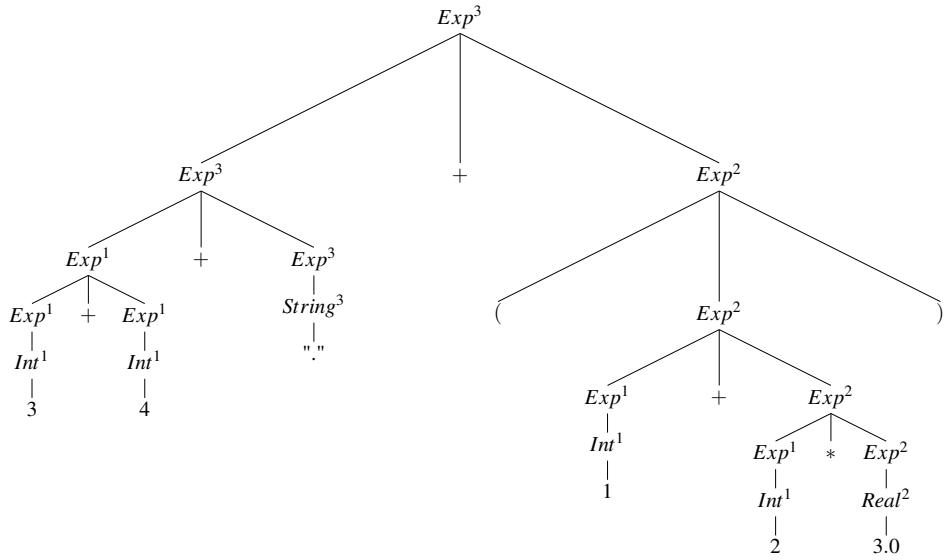
## Solutions

**Ex. 1.** Explain why these methods are (not) semantically correct.

a. No because an object is not a boolean value (but the grammar accept it).

b. Yes because Java auto-wrap basic types to primitive types.

**Ex. 2.** Write the expression tree decorated with the type of each node.



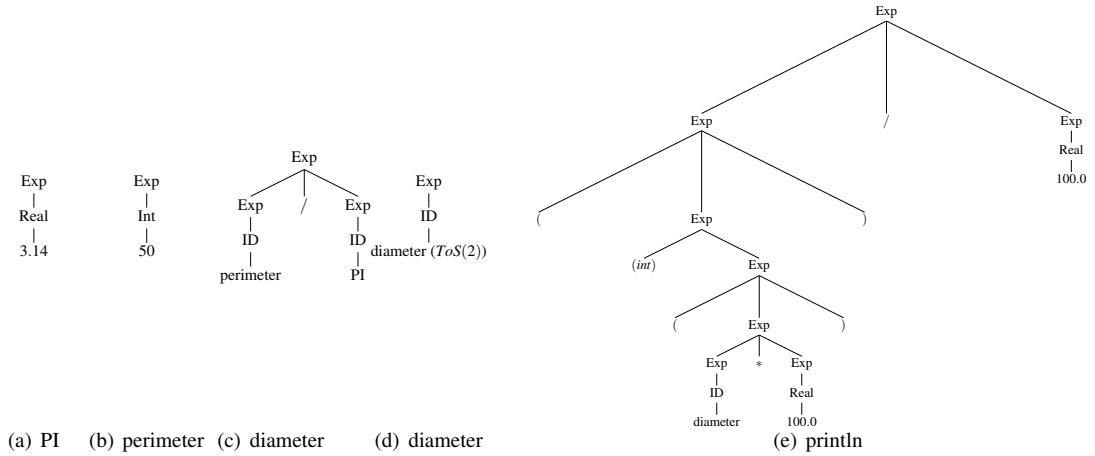

where 1 are **integers**, 2 are **reals** and 3 are **strings**.

**Ex. 3.** Identify the scope of all variables

- Give the table of symbols (ToS)

| UID | Name | Context | Initilization | Type |
|---|---|---|---|---|
| **1** | PI | Exercise3 class | 3.141592653589793 | *double* |
| **2** | diameter | Exercise3 class | / | *double* |
| **3** | args | main function class | parameter | *String[]* |
| **4** | perimeter | main function class | 50 | *double* |
| **5** | diameter | main function | ToS(**2**) | *int* |

- Give the parse tree of each numerical expression




(a) PI (b) perimeter (c) diameter (d) diameter (e) println

- Annotated the parse trees with changes of the table of symbols We add these operations in the root of the trees:

  `PI (a)` $ToS(1) \leftarrow result(Exp)$

  `perimeter (b)` $ToS(4) \leftarrow result(Exp)$

  `diameter (c)` $ToS(2) \leftarrow result(Exp)$

  `diameter (d)` $ToS(5) \leftarrow result(Exp)$

  `println (e) /`

Report any semantic error.

**Line 7: the global `diameter` is a double and the local `diameter` is an integer. A cast operator is required.** This error occurs at the root of the tree (d).