

## Homework 12

### Problem 12.1

#### Solution:

According to Wikipedia: "Some advanced 3GLs like Python, Ruby, and Perl combine some 4GL abilities within a general-purpose 3GL environment. Also, libraries with 4GL-like features have been developed as add-ons for most popular 3GLs. This has blurred the distinction of 4GL and 3GL." I am leaving these languages mentioned above in 3<sup>rd</sup> GL in the table below; above is just a higher level explanation.

Language	Generation	Type	Typing Discipline
C	3 <sup>rd</sup>	Imperative, Von Neumann	Static
C++	3 <sup>rd</sup>	Imperative, Object-Oriented	Static
B	3 <sup>rd</sup>	Imperative	Typeless
Java	3 <sup>rd</sup>	Imperative, Object-Oriented	Static
Python	3 <sup>rd</sup>	Imperative, Object-Oriented, Scripting	Dynamic
Ruby	3 <sup>rd</sup>	Imperative, Object-Oriented, Scripting	Dynamic
Pascal	3 <sup>rd</sup>	Imperative	Static
Basic	3 <sup>rd</sup>	Imperative	Dynamic
Smalltalk	4 <sup>th</sup>	Object-Oriented, Scripting	Dynamic
Perl	3 <sup>rd</sup>	Imperative, Object-Oriented, Scripting	Dynamic
PHP	3 <sup>rd</sup>	Imperative, Object-Oriented, Scripting	Dynamic
Prolog	5 <sup>th</sup>	Declarative	Dynamic

### Problem 12.2

#### Solution:

As the prof has also noted in the question, I am assuming that a `variable` is defined, and below I have written the grammar for reading the conditional expression.

$\langle \text{relational} - \text{oper} \rangle := \text{ } "==" \text{ } | \text{ } "!=" \text{ } | \text{ } "<" \text{ } | \text{ } "<=" \text{ } | \text{ } ">" \text{ } | \text{ } ">=" \text{ }$

$\langle \text{arithmetic} - \text{oper} \rangle := \text{ } "+" \text{ } | \text{ } "-" \text{ } | \text{ } "*" \text{ } | \text{ } "/" \text{ }$

$\langle \text{expr} \rangle := \langle \text{variable} \rangle \text{ } | \text{ } \langle \text{expr} \rangle \langle \text{arithmetic} - \text{oper} \rangle \langle \text{variable} \rangle$

$\langle \text{bool} - \text{expr} \rangle := \langle \text{variable} \rangle \text{ } | \text{ } \langle \text{bool} - \text{expr} \rangle \langle \text{relational} - \text{oper} \rangle \langle \text{variable} \rangle$

$\langle \text{tern} - \text{assign} \rangle := \langle \text{variable} \rangle \text{ } "=" \text{ } \langle \text{bool} - \text{expr} \rangle \text{ } "?" \text{ } \langle \text{expr} \rangle \text{ } ":" \text{ } \langle \text{expr} \rangle$

### Problem 12.3

#### Solution:

From what the question asks, below I am assuming  $\langle identifier \rangle$  and  $\langle constant \rangle$  are already defined and that statements cannot be empty.

$$\langle relational - oper \rangle := "==" \mid "!=" \mid "<" \mid "<=" \mid ">" \mid ">="$$
$$\langle arithmetic - oper \rangle := "+" \mid "-" \mid "*" \mid "/"$$
$$\langle inc/dec \rangle := "++" \mid "--"$$
$$\langle id/con \rangle := \langle identifier \rangle \mid \langle constant \rangle$$
$$\langle bool - expr \rangle := \langle identifier \rangle \langle relational - oper \rangle \langle constant \rangle \mid \langle identifier \rangle \langle relational - oper \rangle \langle identifier \rangle$$
$$\langle while - loop \rangle := "while(" \langle bool - expr \rangle "){" \langle statements \rangle "}"$$
$$\langle statements \rangle := \langle statement \rangle \mid \langle statements \rangle \langle statement \rangle$$
$$\langle statement \rangle := \langle while - loop \rangle \mid \langle identifier \rangle \langle inc/dec \rangle "; " \mid \langle identifier \rangle "=" \langle expr \rangle "; "$$
$$\langle expr \rangle := \langle id/con \rangle \mid \langle expr \rangle \langle arithmetic - oper \rangle \langle id/con \rangle \mid \langle inc/dec \rangle$$