LANGUAGE H: ESPECIFICATIONS, REGULAR EXPRESSIONS AND GRAMMAR

1. INTRODUCTION

There are many reasons to design a programming language. Many people think that designing a programming language takes a tough time. In fact, it is worse, even more if you are a stupid computer science student. The **language H** was designed by Helano Miguel, who is a master in making syntax mistakes while is programming in Java, C or any other programming language. This language was made to avoid certain types of syntax mistakes. However, given the skill level of the designer (who sucks in any programming language), this language has 99.9% of chances to fail. Helano thinks that he is just a terrible programmer, but the truth is that he is a terrible English writer as well. Enjoy a great time trying to not find a mistake in this document. Let’s get started.

2. RESERVED WORDS

- Int
- double
- text
- for
- void
- return
- if
- else
- switch
- case
- break
- default
- boolean
- and
- or

3. OPERATORS

- Arithmetic: + , - , *, /.
- Attribution: =
- Boolean logic: and, or , !.
- Comparison: ==, !=, >, <, <=, >=.

4. CHARACTERS

- Lower case and uppercase letters: a – z , A – Z
- Decimal digits: 0 – 9
- Graphic Characters: " # % & ' ( ) * + , - . / : ; < = > ? [\ ] ^ _ { | } ~

5. REGULAR EXPRESSIONS:

Int : [I.n.t]

double: [d.o.u.b.l.e]

text: [t.e.x.t]

for: [f.o.r]

void: [v.o.i.d]

return: [r.e.t.u.r.n]

if: [i.f]

else: [e.l.s.e]

switch: [s.w.i.t.c.h]

case: [c.a.s.e]

break: [b.r.e.a.k]

default: [d.e.f.a.u.l.t]

boolean: [b.o.o.l.e.a.n]

and: [a.n.d]

or: [o.r]

5.1 REGULAR EXPRESSION OF IDENTIFIERS:

- **Variables:** [V] [a-z | A-Z | 0-9 | _] *
- **Functions:** [F] [a-z | A-Z | 0-9 | _] *

Comments:

Line comment: [/./][a-z | A-Z | 0-9 | # % & ' ( ) * + , - . / : ; < = > ? [\ ] ^ _ { | } ~]*

Block comment: [/.*].[a-z | A-Z | 0-9 | # % & ' ( ) * + , - . / : ; < = > ? [\ ] ^ _ { | } ~]* [*./]

Integer numbers: [0-9]+

Double numbers: ([0-9]+“.”[0-9]*)|([0-9]*“.”[0-9]+)| [0-9]+

Text data: ([a-z] | [A-Z])+

Nothing to do: (“ ” | “\n” | “\t”)+