The Converter Grammar

Here is the syntax-only version of the grammar on which my solution is based. I draw your attention to a few points:

A criteria file can mention any of the years any number of times, in any order. The examples in the kit were, perhaps a bit misleading, because they might have implied that `AnyYear` was mandatory. Of course one could have written a grammar with that in mind, but I didn't want that.

The `Expression` hierarchy is essentially the same as the one in your Parva compiler kit, but the complications/features to support a character type have been removed (no casting, for example).

So the whole grammar is in fact pretty simple, part of it reminiscent of your Parva Compiler and part of it reminiscent of the "meet the family" prac question.

In case you missed it, `IGNORECASE` means that the few key words like `FirstYear` could be written in any combination of upper and lower case. `IGNORECASE` does not reduce identifiers and strings to a consistent uppercase, however, you have to do that in the parser if you want the case-insensitive feature. I suSPect the DeAN of huMANities WOulD appreciate that.

```
COMPILER Converter  $CN
/* Converter for generating a C# method for transforming a list of criteria for marking
   up student exam results from a simple specification into equivalent C# code.
   Context free grammar only.  Exam Kit provides an attributed version of this
   P.D. Terry, Rhodes University, 2013 */

IGNORECASE

CHARACTERS
  letter       = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" .
  digit        = "0123456789" .

TOKENS
  integer      = digit { digit } | digit { digit } CONTEXT (".") .
  double       = digit { digit } "." digit { digit }  .
  identifier   = letter { letter | digit | "_" ( letter | digit ) } .

IGNORE CHR(0) .. CHR(31)

PRODUCTIONS
  Converter    = Faculty
                 {   "FirstYear"  ":" CriteriaList
                   | "SecondYear" ":" CriteriaList
                   | "ThirdYear"  ":" CriteriaList
                   | "FourthYear" ":" CriteriaList
                   | "AnyYear"    ":" CriteriaList
                 } .

  Faculty      = Identifier .

  CriteriaList = { Action ":" Condition { ";" Condition } "." } .
  Action       = Identifier .

  Condition    = Expression .
  Expression   = AndExp { OrOp  AndExp } .
  AndExp       = EqlExp { AndOp EqlExp } .
  EqlExp       = RelExp { EqlOp RelExp } .
  RelExp       = AddExp [ RelOp AddExp ] .
  AddExp       = MulExp { AddOp MulExp } .
  MulExp       = Factor { MulOp Factor } .
  Factor       = Primary | ( "+" | "-" | NotOp ) Factor .
  Primary      = Designator | Double | Integer | "(" Expression ")" .
  Designator   = Identifier .

  Identifier   = identifier .
  Integer      = integer .
  Double       = double .

  AddOp        = "+" | "-" .
  MulOp        = "*" | "/" .
  EqlOp        = "=" | "/=" .
  RelOp        = "<" | "<=" | ">" | ">=" .
  NotOp        = "not" .
  AndOp        = "and" .
  OrOp         = "or" .

END Converter.
```