The DFile PL (Erwin’s slide)

**Historical Background:**
- MS Access, SQL Server, Oracle
- Conversion Wizards

**DFile**
- Motivation
- Ease of use, database functionality without the complicated interfaces
- HTML reports
Welcome to DFile (Amrita’s 1st slide)

- Language name: DFile
- Compiler name: DFiler
- Command line input: yes
slide_title = grammar;

- All standard operators...and...
  - & (Concatenation)
    - “Hello” & “there” = “Hello there”
    - 3.0 + 4.0 = 7.0, but 3.0 & 4.0 = “3.04.0”
  - ~= (outer scope reference)

- Control flow constructs
  - If-else, for, foreach (2 variations)

- Functions: user-defined and built-in

- Libraries for report generation
...less is more...

```cpp
#include "reportsLib.df"

generateReport("sampledata");
```
...yet another slide by Amrita

include "reportsLib.df";

heading = "this is the heading";
footer = "this is the footer";
colnames = ["col0", "col1", "col2", "col3"]; 

function where()
{
    return = #3 > 0;
}

generateReport("sampledata");
yet another slide by Amrita

include "reportsLib.df";

heading = "this is the heading";
footer = "this is the footer";
colnames = ["col0", "col1", "col2", "col3"];

function where()
{
    return #3 > 0;
}
generateReport("sampledata");
Architecture (Howie’s turn)

Front End

Lexer

Parser

Walker

Read input and break it into TOKENS

Take TOKENS and check for errors. Create the AST.

Walk AST evaluating TOKENS. Calls functions as required.

Back End

Calls functions as required

Helper Functions

Activation Records

Symbol Tables

Exception Handling

File I/O
Howie’s other slide

- Activation Records
- Error reporting with Line Numbers
- Foreach – disambiguating the structure.
Summary (now everyone talk)

- Successfully built DFile.
- DFile evolved to a generic yet powerful PL rather than just a report generator.
Lessons Learnt (confession time)

- Learnt the inner workings of a compiler.
- Old project sources are GOLD!
- Lecture Notes are significant sources during implementation