MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

159.341 Programming Languages, Algorithms & Concurrency

Programming Languages - Part 4

Daniel Playne
d.p.playne@massey.ac.nz

Language Evaluation

We have looked at a number of different ways to classify different programming languages, which is useful when trying to understand a language.

Consider different ways to compare and evaluate languages (from Sebesta)

- Readability
- Writability
- Reliability

Readability

Readability is one of the most important criteria for judging a programming language. It represents the ease with which a program written in a certain language can be read and understood.

As machines and program size increase, emphasis moves from efficiency and more towards maintainability. This represents a crossover from ***machine orientation*** to ***human orientation***.

The maintainability of a program is largely determined by how readable a program is.

Readability

There are several characteristics of a language that contribute to the overall readability of a language - ***simplicity***, ***orthogonality***, ***types*** and ***syntax***.

Simplicity - the overall simplicity of a language strongly affects its readability. A language that consists of a large number of constructs will be more difficult to learn and inevitably mean that programmers only really use a specific subset. This means the writer and the reader may be familiar with different subsets.

Readability - Simplicity

Another factor that contributes to simplicity is ***feature multiplicity*** - languages providing multiple ways to accomplish the same thing. For example all of the following code example increment the value of an integer by one:

```
count = count + 1;
count += 1;
count++;
++count;
```

Readability - Simplicity

Operator overloading can also affect the simplicity of a language. Allowing a user to define operators to perform any function means that no consistency is enforced on the meaning of an operator.

Simplicity can be taken too far, after all assembly languages consist of simple constructs but lack complex control structures which makes program structure less obvious (and less readable).

Readability - Orthogonality

Orthogonality - in a programming language orthogonality means that a relatively small set of constructs can combined to form the data and control structures of the language.

A language feature is orthogonal if its meaning is independent of the context in which it appears.

The word orthogonal comes from mathematics and orthogonal vectors which are independent of each other.

Readability - Orthogonality

C is not orthogonal in its treatment of many concepts and language structures.

For example, it has two main kinds of structured data - arrays and records (structs). A structure can be returned from a function but an array cannot except if there is an array inside a structure then it can be.

Parameters are passed to functions by value, except for arrays.

A variable cannot be declared of type `void`, but can be with type `void*`.

Readability - Types

Data types - the treatment of data types and data structures can have an important effect on the readability of a program. Consider a language that does not support boolean types and numeric values are used instead.

```
timeOut = 1
```

The meaning of this statement is unclear as there is nothing particular about it that tells a reader that this is being used as a boolean condition. The following is clearer.

```
timeOut = true
```

Readability - Syntax

Syntax design - the syntax of the elements of a language has a significant effect on the readability of programs. The choice of language keywords is important, as is the method of grouping statements.

Languages (such as C) that close different compound statements with the same symbol } can cause confusion as it is not always clear which group is being closed.

Some languages (such as Ada) use different syntax to close different groups `end if`, `end loop` etc.

Writability

Writability represents how easily it is to write a program in a given language.

Naturally there is a significant overlap between readability and writability - programmers spend a lot of time reading their code while they write it.

Characteristics - ***simplicity***, ***orthogonality***, ***abstraction***, ***expressivity***

Writability

Simplicity and ***Orthogonality*** - languages with a large number of features or features that are not orthogonal can lead to the misuse of features, inelegant solutions or even accidental usage features.

A language that provides a smaller number of constructs with a consistent set of rules can allow programmers to easily learn a language and design solutions.

Writability - Abstraction

Abstraction Support - abstraction is the ability to define structures or operations that can be used while hiding many of the details.

Support of abstraction has a strong impact on the design of a program and thus the writability of a language.

Writability - Expressivity

Expressivity - the expressivity of a language generally refers to convenient it is to express computations. The expressivity of a language can also mean a large amount of computation can be defined with a relatively small amount of code.

For example, array notation can allow a large amount of computation to be defined with a simple expression.

```
A[:]  = B[:]  + C[:]
for(int i = 0; i < 10; i++) {A[i] = B[i] + C[i];}
```

Reliability

A program can be said to be reliable if it always performs to the specifications under any and all conditions.

Languages may provide features that make it easier/harder to write reliable programs (e.g. Rust).

The following features of a language can help in developing reliable programs: ***type-checking***, ***exception handling***, ***aliasing***

Reliability - Type Checking

Type-checking - as we have already discussed type-checking is testing for errors in a program, either during compilation or run-time.

This is an important factor in the reliability of a program. Failure to check type-properly can easily lead to many errors.

In the original C language, the types of procedure parameters where not type-checked and led to countless errors.

Reliability - Exceptions

Exception-Handling - is the ability of a program to detect and intercept run-time errors or other unusual conditions and respond to them in an appropriate manner.

These facilities are obviously important for program reliability and are present in most widely-used languages - Ada, C++, C#, Java but noticeably absent from languages like C and FORTRAN.

Reliability - Aliasing

Aliasing - loosely defined aliasing is having two or more separate names that are used to address to the same variable in memory.

Aliasing can easily cause problems in many programs (double-delete, accessing deleted memory etc). Changing the value of one variable affects the value of another (or more). This can easily cause problems related to reliability and thus many languages try to restrict aliasing.

Reliability

Readability and ***writability*** - both the previous criteria strongly affect the reliability of programs written in a given language.

The ease with which programmers can design and write programs and then subsequently read them and analyse them inevitably affects the reliability of the program.

Languages that require programs to be written in unnatural approaches make it hard for programmers to identify potential problems.

Language Evaluation

Characteristic	Readability	Writability	Reliability
Simplicity	•	•	•
Orthogonality	•	•	•
Data Types	•	•	•
Syntax	•	•	•
Abstraction		•	•
Expressivity		•	•
Type Checking			•
Exception Handling			•
Restricted Aliasing			•

Summary

- Language Evaluation
- Readability
- Writability
- Reliability