

Lecture 04

Language Evaluation Criterion

Mr. Mubashir Ali

Lecturer (Dept. of Computer Science)

dr.mubashirali1@gmail.com

Summary of Previous Lecture

- 1. Capers Jones Table**
- 2. Hello World (BASIC, LISP, C, C++)**
- 3. Language Evaluation Criterion**

Outline

- 1. Language Evaluation Criterion Continued**
- 2. Readability**
- 3. Writability**
- 4. Orthogonality**
- 5. Reliability**

Readability - Control Statements

- Hazards of goto
- Restricted use was needed and useful as well
- Probably not an issue any more

Readability - Syntax

- Variable names
- Special keywords for signaling the start and end of key words.

```
if (some condition)
  do this
  now do this
```

```
if (some condition) then
  do this
end if
  now do this
```

Writability

- Support for abstraction
- Expressivity
 - Constructs that make it convenient to write programs e.g. for set data type in Pascal we do not have an equivalent in C

Orthogonality

- Relatively small number of components that can be combined in a relatively small number of ways to get the desired results.
- Closely associated with simplicity
- The more orthogonal the design the fewer exceptions
 - makes it easier to learn, read, and write
- The meaning of an orthogonal feature is independent of the context – symmetry and consistency.
 - e.g. pointers

Orthogonality

IBM Mainframe

A	Reg1, memory_cell
AR	Reg1, Reg2

VAX

ADDL	operand1, operand2
------	--------------------

Orthogonality

- C
 - You can return structures but not arrays
 - An array can be returned if it is inside a structure
 - A member of a structure can be any data type except void or the structure of the same type
 - An array element can be any data type except void
 - Everything is passed by value except arrays
 - Pointer arithmetic
 - $a+5$ means what?
 - $a + b$
 - Use of void as a type in a structure

Orthogonality

The other side...

- In C – all statements return some value and hence can be used in expressions.
 - Logic and arithmetic expressions can be intermixed
 - Can cause side effects
 - Can cause cryptic code
- $A[i]$, $i[A]$
- Since languages need large number of components, too much orthogonality can cause problems.

Reliability

- Type Checking
 - Run-time checking versus compile time checking
 - Mismatched parameters
 - Array bounds
- Exception handling
 - intercept run-time errors and take corrective measure if possible.

Hello World Programs

from: Infiltec Humor Page
www.infiltec.com

High School/Jr.High – BASIC
2 lines

```
10 PRINT "HELLO WORLD"  
20 END
```

Senior year in College - LISP

3 lines

```
(defun hello  
  (print  
    (cons 'Hello (list 'World))))
```

New professional – C

10 lines

```
#include
void main(void)
{
    char *message[ ] = {"Hello ", "World"};
    int i;
    for(i = 0; i < 2; ++i)
        printf("%s", message[i]);
    printf("\n");
}
```

Seasoned professional – C++

```
#include
```

```
#include
```

```
int main()
```

```
{
```

```
    String str;
```

```
    str = "Hello World";
```

```
    cout << str << endl;
```

```
    return(0);
```

```
}
```

Language Evaluation Criteria

I have reaffirmed a long-standing and strongly held view: Language comparisons are rarely meaningful and even less often fair. A good comparison of major programming languages requires more effort than most people are willing to spend, experience in a wide range of application areas, a rigid maintenance of a detached and impartial point of view, and a sense of fairness.

Bjarne Stroustrup

Language Evaluation Criterion

- How to compare and examine languages?
- Base Criteria:
 - Readability
 - Writability
 - Reliability

Readability

- Directly related to the cost of maintenance
- Choice of language for a particular task
 - Simplicity
 - Orthogonality
 - Control Statements
 - Data types and data structures
 - Syntax Considerations

Readability - Simplicity

- Number of basic components
 - Learning curve
 - Subset can be used for writing but for reading you must know everything
- Feature multiplicity – more than one way to accomplish the same task.
 - `i = i + 1; i++; ++i; i += 1;`
- Operator overloading
 - problematic if inconsistent or unconventional
- How much simple should it be?

Summary

- 1. Language Evaluation Criterion Continued**
- 2. Readability**
- 3. Writability**
- 4. Orthogonality**
- 5. Reliability**