



PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria

Prafullata Kiran Auradkar

Computer Science and Engineering

PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria

Prafullata Kiran Auradkar
Computer Science and Engineering

PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria



1. Readability
2. Writability
3. Reliability
4. Cost

PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria

T1: Table 1.1 Language evaluation criteria and the characteristics that affect them.

Characteristic	CRITERIA		
	READABILITY	WRITABILITY	RELIABILITY
Simplicity/orthogonality	•	•	•
Control structures	•	•	•
Data types and structures	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability - Overall Simplicity



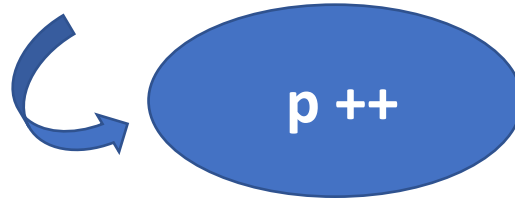
- Number of basic constructs
 - Too many constructs will complicate the usage.
 - Tendency of learning/using a subset.
- Number of basic constructs
 - Too many constructs will complicate the usage.
 - Tendency of learning/using a subset.
- Feature multiplicity – to be limited
 - Shortcuts can make it easier to use, but readable?
- Operator overloading
 - Programmer can change/add the meaning of the existing operator. ($S1 + S2 \rightarrow$ results in what?)
 - Leads to confusion...
- Simplicity in Assembly level languages
 - Program structure is difficult to read

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability - Orthogonality

- Fewer exceptions in the language
- Implies simplicity and regularity in language design
- Language feature independent of the context of its appearance in the program

Simple example:



Data types – int, float

Operators - + and /

Valid combinations for +

- Int + Int
- Float + Float
- Int + Float
- Float + Int

Valid combinations for /

- Float / Int
- Float / Float
- Int / Int

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability - Orthogonality



- Data types in **C** – are they Orthogonal or Not?
 - Pointers
 - Structured types
 - Structures
 - Arrays

Caution: Too much Orthogonality adds to the language complexity

Ex: ALGOL

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability – Control Statements

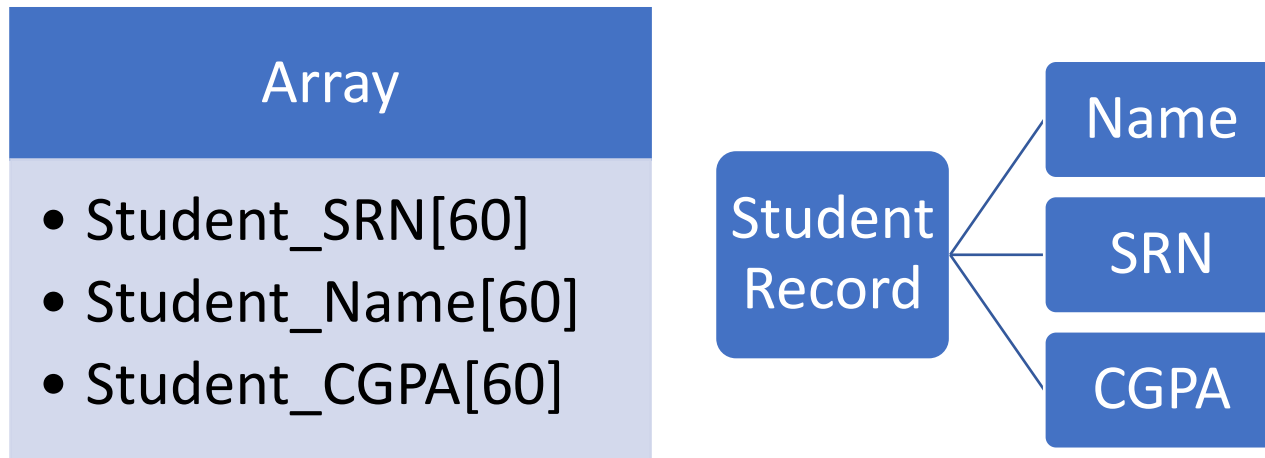
- Well defined control structures – While, Do-While, For, Repeat-Until...
- GoTo... ?

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability – Data types and Data Structures



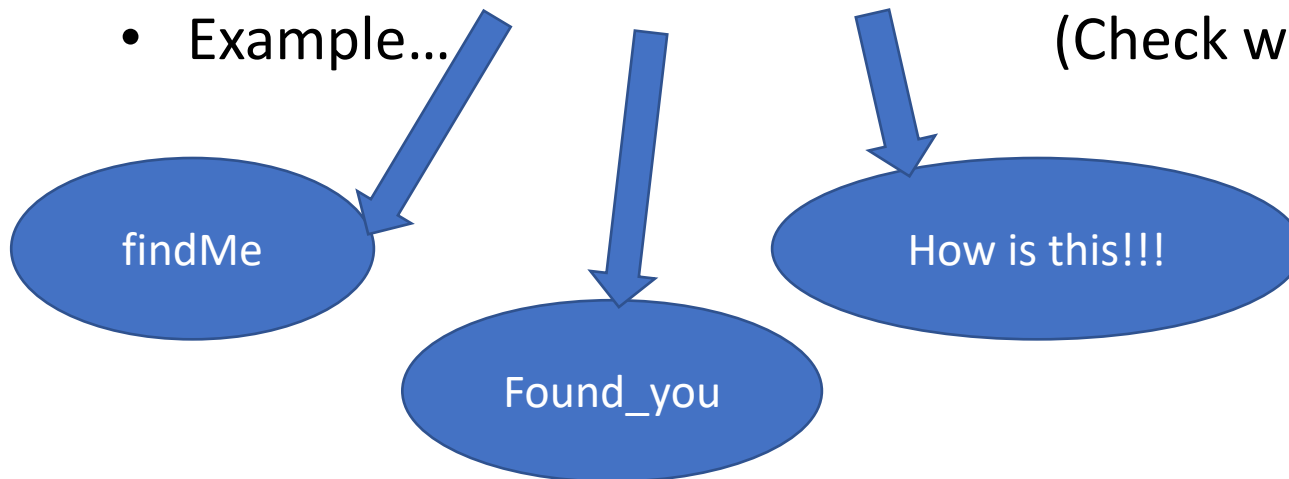
- Support of data types – for better readability
 - Ex: Represent TRUE/FALSE as Boolean type or integer type?
 - Record types or Arrays?



PRINCIPLES OF PROGRAMMING LANGUAGES

Readability – Syntax Design

- Identifier forms
 - Special words
 - If keywords can be used as variables, it leads to confusion.
 - 'real' and 'End' can be used as variable names in Fortran!!!
 - How do we connect two words?
 - Example...
- (Check with first.f)



PRINCIPLES OF PROGRAMMING LANGUAGES

Readability – Syntax Design

Compound statements

Curly braces	Distinct closing words	indentation
Class abc { Method () { If(...) { ... } } }	procedure testtype.. Begin ... end testtype; End if Or end loop	def findavg(): ... while myList: total += 20 avg=...

PRINCIPLES OF PROGRAMMING LANGUAGES

Readability – Syntax Design

- Form and meaning
 - Self-descriptive constructs and meaningful keywords
 - Meaning of 'static' in C depends on the context.
- UNIX shell command. - The command ***grep*** is not obvious.



PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria

T1: Table 1.1 Language evaluation criteria and the characteristics that affect them.

Characteristic	CRITERIA		
	READABILITY	WRITABILITY	RELIABILITY
Simplicity/orthogonality	•	•	•
Control structures	•	•	•
Data types and structures	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

PRINCIPLES OF PROGRAMMING LANGUAGES

Writability

- Simplicity and Orthogonality
 - Few constructs, a small number of primitives, a small set of rules for combining them
 - Errors can go undetected when there is a huge set of constructs available.
- Expressivity
 - A set of relatively convenient ways of specifying operations
 - Example: the inclusion of for statement in many modern languages

PRINCIPLES OF PROGRAMMING LANGUAGES

Writability

- Support for abstraction
 - The ability to define and use complex structures or operations in ways that allow details to be ignored
 - Data abstraction
 - Classes and Objects
 - Building trees.
 - Process abstraction
 - Modular approach through subprograms

PRINCIPLES OF PROGRAMMING LANGUAGES

Language Evaluation Criteria

T1: Table 1.1 Language evaluation criteria and the characteristics that affect them.

Characteristic	CRITERIA		
	READABILITY	WRITABILITY	RELIABILITY
Simplicity/orthogonality	•	•	•
Control structures	•	•	•
Data types and structures	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

PRINCIPLES OF PROGRAMMING LANGUAGES

Reliability

- Type checking
 - Testing for type errors
 - Run time Vs. compile time
- Exception handling
 - The ability of a program to intercept run-time errors and continue after taking corrective measures.
- Ada, C++, Java, Python support exception handling whereas C does not.



PRINCIPLES OF PROGRAMMING LANGUAGES

Reliability

- Aliasing
- Readability and writability
 - Programs that are difficult to read are difficult to write (and also to modify).
 - Affects reliability

PRINCIPLES OF PROGRAMMING LANGUAGES

Cost

- Training programmers to use a language
- Compiling programs
- Executing programs
- Language implementation system:
 - availability of free compilers
- Reliability: poor reliability leads to high costs
- Maintaining programs



PRINCIPLES OF PROGRAMMING LANGUAGES

Additional Criteria



- Portability
 - The ease with which programs can be moved from one implementation to another
- Generality
 - The applicability to a wide range of applications
- Well-definedness
 - The completeness and precision of the language's official definition



THANK YOU

Prafullata Kiran Auradkar
Computer Science and Engineering
prafullatak@pes.edu