PRINCIPLES OF PROGRAMMING LANGUAGES

Reasons for studying concepts of programming languages

- To improve your ability to develop effective algorithms.
- To improve the use of existing programming languages
- To allow a better choice of programming language
- To make it easier to learn a new language.
- To make it easier to design a new language.

Programming Domains

1. Scientific applications.
2. Business applications
3. Artificial intelligence.
4. System programming.
5. Special purpose languages.

Language Evaluation Criteria

4 characteristics.

1. Readability - for software maintainance
2. Writability
3. Reliability.
4. Cost

Readability - The ease by which you can read and understand the program.

a) Overall simplicity - a language having less no of features are better than languages having large number of features.

b) Feature multiplicity - Less feature multiplicity is the better. eg: count = count +1, count += 1, count++

c) Operator Overloading - Reduces readability.

c = a + b.

d) Orthogonality (simplicity)

IBM

A Reg l , memory cell

Reg l ← content (Reg 1)+ content (memory cell)

AR Reg l , Reg 2

Reg 1 ← content (Reg 1) + content (Reg 2) .

VAX

ADDL operator 1, operator 2

operand 2 ← content (operator 1)+ content (operator 2).

e) Control Statements .

To make the program more readable,

- Reduce the control statements .
- goto statement and target statement should be closer to each othere .
- goto statement should precede the target statement .

Scanned by CamScanner

f) Datatypes and Structures .

Appropriate datatypes and structures should be there for better readability .

g) Syntax Consideration .

Syntax : The way in which the program appears.

i) Identifier forms .

Identifier should have enough length to convey the meaning .

eg: stud_name for student name .

ii) Special words .

Special words is generally not used as variable . It is used for a specific purpose . Eg : In C , for is used for loop statement .

iii) Form and Meaning .

In terms of readability, one meaning for a form is better .

Writability.

The ease with which we can write a program.

Factors influencing writability,

1. Simplicity and Orthogonality.

   Language having less number of features are better.

2. Support for Abstraction.

   Two types of abstraction - Process abstraction.
   - Data abstraction.

   These features increases writability.

3. Expressivity.

   How easily we can express / write a program

   count++ is more expressive than count+=1 or

● Reliability.

Under all circumstances, a program should give expected output.

i) Type checking.

Checking that each operator receives

proper number of arguments of proper data type.

Compile time type checking - Eg: C, C++, Java

Run time type checking - Eg: Python, Ada

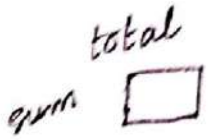
2. Exception handling.

To handle run time errors.

Eg: Division by 0.

3. Aliasing.

Different names are given for the same memory cell.



4. Readability and Writability.

Languages having less no. of features are more reliable.

Cost.

Cost involved in training programmers.

Cost of writing the programme.

Cost involved in compiling the programme.

Cost of executing the programme.

Cost involved due to poor reliability.

Cost of maintaining the programme.

Describing Syntax and Semantics.

Syntax - The form of expressions, statements and program units is called syntax of a programming language.

Semantics - The meaning of expressions, statements and program unit is called semantics of programming language.

Lexeme - Lowest syntatic unit of a programming language.

Eg: a = b + c * 2.   a, =, b ... all are lexeme

Token - These lexemes belongs to some category, that category is called token.

Eg: a, b, c etc belong to the token 'identifier'.

+, -, * etc belong to the token 'operator'.

Grammar - The notation used to specify the syntax of a programming language.

A scientist named Chomsky classified grammar into 4 types. They are:

- Type 0
- Type 1
- Type 2 Syntax of PL Context free grammar (BNF
  Eg: $a = b\boxed{+*}c$.
  invali
- Type 3 Regular grammar
  Eg: $\boxed{2a} = b + c$.
  invalid.

BNF - John Backus & Peter Nawns Form.

<u>Quantities Involved in Grammar</u>.

4 quantities are involved in grammar. They are:

i) Terminal : The basic unit of a PL.

ii) Non-terminal : denotes set of strings (can be further expanded)

iii) Start Symbol : One of the non terminal is selected as the start symbol.

iv) Production : Rule.