UNIT 1

Vaibhav Khatavkar

College of Engineering, Pune

January 4, 2019

1 Programming Domain

Programming Domains

"Influence of Computers over mankind!!!"

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

- What is computer ?

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

- What is computer ? I/P , Process , O/P

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

- What is computer ? I/P , Process , O/P
- For processing, computer uses its own instruction set. To be precise machine instructions.

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

- What is computer ? I/P , Process , O/P
- For processing, computer uses its own instruction set. To be precise machine instructions.
- Machine Instructions are in machine level language of 0's and 1's. Its difficult to read and understand them. Ex. A code of rocket launcher at NASA was written in machine language. The rocket launching failed cz of a single 1 instead of 0 in huge code

Programming Domains

"Influence of Computers over mankind!!!"

Broad Domains :

- Scientific Applications : FP arithmetic computations, efficiency (1940s ALP) [FORTRAN,ALGOL]
- Business Applications : reports, store decimals and chars [COBOL]
- Artificial Intelligence : Symbolic computations rather than numbers [LISP, PROLOG]
- Systems Programming : OS [C]
- Web Software : www [HTML,XHTML, CSS , JS , PHP ....]

Lets think on ...

- What is computer ? I/P , Process , O/P
- For processing, computer uses its own instruction set. To be precise machine instructions.
- Machine Instructions are in machine level language of 0's and 1's. Its difficult to read and understand them. Ex. A code of rocket launcher at NASA was written in machine language. The rocket launching failed cz of a single 1 instead of 0 in huge code

Role of programming languages

Initial Goal : to execute programs efficiently.

- In olden days (Mid 60s), computer was a critical resource and programmers were not expensive. Languages : FORTRAN, COBOL,LISP and ALGOL. Main task was to compile programs on a large expensive computer
- After few days, machines were less expensive. Programming , porting, maintenance cost was larger than computer cost. Main task was to make it easier to develop correct programs to solve problems for some given application area.
- During 1960 to 1970 compiler technology matured. Main task was to solve domain specific problem. Eg. For scientific applications - FORTRAN, business applications - COBOL, Military applications - JOVIAL , AI - LISP, embedded military applications - Ada.
- Programming Languages evolved. ALGOL was replaced by Pascal, which in turn was replaced by C++ and Java.COBOL replaced by C++. Even languages like APL,PL/I and SNOBOL4, Pascal disappeared.

A programming language

- Why a particular language is popular/ dead ?

A programming language

- Why a particular language is popular/ dead ?

ANS Each language has Pros and cons.
There may be external reason.
e.g. use of COBOL or Ada was enforced in US by Govt.
FORTRAN - strong support by manufacturers

A programming language

- Why a particular language is popular/ dead ?

ANS Each language has Pros and cons.
There may be external reason.
e.g. use of COBOL or Ada was enforced in US by Govt.
FORTRAN - strong support by manufacturers

- Why a programmer prefers 'X' language over 'Y' ?

A programming language

- Why a particular language is popular/ dead ?

ANS Each language has Pros and cons.
There may be external reason.
e.g. use of COBOL or Ada was enforced in US by Govt.
FORTRAN - strong support by manufacturers

- Why a programmer prefers 'X' language over 'Y' ?

ANS Syntax and Sematics, programming environment, ...

A programming language

- Why a particular language is popular/ dead ?

ANS Each language has Pros and cons.
There may be external reason.
e.g. use of COBOL or Ada was enforced in US by Govt.
FORTRAN - strong support by manufacturers

- Why a programmer prefers 'X' language over 'Y' ?

ANS Syntax and Sematics, programming environment, ...

Lets see some more reasons...

Attributes of a good language

Clarity, simplicity and unity Conceptual integrity, semantic differences should reflect in syntax.

Orthogonality Features when combined in various ways should be meaningful.

Naturalness of application syntax $->$ program structure $->$ logical structure.

Support for abstraction Allow data structures, data types and operations to be defined and maintained as self-contained abstractions.

Program verification Simplicity of semantic and syntactic structure

Programming environment

Portability of programs

Cost of use Cost of program execution, translation, maintenance creation, testing and use.

Why to study concepts of programming languages ?

Increases capacity to express ideas

Improves background for choosing appropriate language

Increases ability to learn new languages

Better understanding of the significance of implementation

Programming Paradigms

Many people debate on the efficiency of the array declaration in C++ versus Java or value interpretation vs compiling program.
We should focus on understanding how languages are constructed.
Some definitions : [2]

Programming paradigm is an approach to programming a computer based on a mathematical theory or a coherent set of principles.

Programming language is a tool for developing executable models for a class of problem domain

Four basic computational models :

- Imperative / procedural : command driven/ statement oriented
- Applicative / functional : functions
- Rule based/ logical : Checking presence of certain conditions and then excuting appropriate action
- Object Oriented : Objects are built and limited set of functionalities is provided to operated those data.

[2] https://sites.google.com/site/cs4217jan2011team2/resources

- Declarative
  - functional: LISP/Scheme, ML, Haskell
  - dataflow: Id,Val
  - logic,constraint-based: Prolog, spreadsheets
  - template-based: XSLT
- Imperative
  - von Neuman: C, Ada, FORTRAN....
  - Scripting: Perl, python, PHP....
  - Object-oriented: Smalltalk, Eiffel ,Java

There are many languages developed.
Jan 18 Top-10 languages are :Java, C , C++ , Python, C#, JS, VB.net, R, PHP, Perl
[3]

[3] https://www.tiobe.com/tiobe-index/

Why to study Programming paradigms

1. Increasing no. of programming languages implementing similar paradigms exists.
2. 27 paradigms in total but some are in similar concept
3. Studying the 4 distinct basic programming paradigms allow us to easily pick up any programming language.

Structured Sequence Control

Data Types

What does a programming language consists ?

Data Types

What does a programming language consists ?

- Data types

Data Types

What does a programming language consists ?

- Data types
- Operators

Data Types

What does a programming language consists ?

- Data types
- Operators
- Statements

Data Types

What does a programming language consists ?

- Data types
- Operators
- Statements
- Procedures

Data Types

What does a programming language consists ?

- Data types
- Operators
- Statements
- Procedures

and many more .....

Data Types

- Structured Datatype : The manner in which data items are represented, which may be same or different scalar data types. eg array
- Unstructured Datatype : Each data item require different data types and have different characteristics . eg audio files , images , video files , mails , text files, etc.

Data type

- Computer hardware interprets bits in memory in diff. ways : as instructions , addresses , characters, integers and floating point numbers
- It does not attempt to keep track of which interpretations correspond to which locations in memory. “Bits are untyped”
- Even in assembly you can perform operations of any kind with arbitrary locations.
- High-level languages associate types with values.
- It provides contextual information and error checking

Typed System

Definition

consists of

1. a mechanism to define types and associate them with certain language constructs
2. a set of rules for *type equivalence, type compatibility, and type inference*

constructs : constants, variables , records fields , parameters, subroutines, expressions
type equivalence : types of two values are same
type compatibility : value of type can used in given context

type inference : type of expression based on constituent parts

Type Checking

Definition (Type checking)

is a process of ensuring that a program obeys the language's type compatibility rules.

Definition (Strongly typed language)

if it prohibits application of any operation to any object that is not intended to support that operation.

Definition (Statically typed language)

if it is strongly typed and type checking is performed at compile time,

Definition (Dynamically typed language)

type checking done at run-time i.e. late binding.