# Survey of Programming languages

Categories of application domain

* Scientific application
* Data processing application
* Text processing application
* Artificial intelligence application (works with historical data to interact with the machine)
* System programing application (OS): it uses a kernel
* Web programming apps

Not every application works with every program because they are tailored to satisfy certain problem

Programming Language Evaluation Criteria

1. Expressivity: It is the ability of a language to clearly express the
2. Well refinedness: In every language there is the syntax and the semantics. It should be able to take care of errors that come out of both.
3. Data type and structure:
4. Readability:
5. Modularity: It must contain Modules (the problems solved): ability to solve problems
6. Input/output Facilities:
7. Portability: Transferable from one device to another
8. Efficiency: How fast is it processing and how much memory is it using? (Speed and space consumed)
9. Generality: Must be able to work without restriction.

Language methodology

It is the approach to software planning and the process to software development.

Language Paradigms

The style of writing computer programs

Programming paradigms: Why have different styles of writing programs?

* Advancement in computer technology (Software and Hardware).
* Human creativity
* Imperative Programming: Some of the earlier programming languages that were used. It was mostly written in top to bottom. It gives a step by step representation of how the programming should be done.
* Procedural Programming: It allows programming to be developed in a structural approach instead of a step by step. It is written in routines.
* Functional Programming; It is rooted in mathematics and deals with function. It has its roots in math and it is language dependent. E.g. Python, MIPS, etc.
* Declarative Programming: It is a style of programming that deals with a special kind of logic. e.g. SQL.
* Object-Oriented Programming: Everything is seen as a collection of class or object meant for communication. More emphasis is on data rather than
* Scripting Languages: it is a fast tracking representation of the English language.

**Programming Approaches**

- Down Up (Bottom Up): The separate parts join together to form a larger entity.

- Top down: Each unit is broken down to a module.

Language Trade-Off Design

Needs to satisfy

* Programmer -> Productivity
* Machine -> Performance
* Us#
* Reliability: time required for malfunction and error detection and handling
* Expandability: Ability to accommodate without a need for
* Maintainability:
* Compatibility:
* Adaptability:
* Availability:
* Cost:

# 