

Name : Rohit Kumar

Course : CCA

Assignment: Introduction to programming language.

Date : 09-08-2021

Introduction to Programming Languages

Special Features of the text

The following are significant features of the text as compared to the standard texts

- Syntax! an introduction to regular expression, context-free grammars, parsing, attribute grammars and abstract grammars
- Semantics! introductory treatment of algebraic, axiomatic, denotational and operational semantics.
- Programming paradigms! the major programming paradigms are prominently featured.



- Languages design principal. Intendly some programming language design principle are given prominence.

Readership:-

This book is intended as an undergraduate text in the theory of programming languages. To gain maximum benefits from the text, the reader have should experience in a high-level programming languages such as pascal, Modula-2, C++, ML or Common Lisp, machine organization and programming and discrete mathematics.

Page No. 3

Programming is a spectator sport. To gain maximum benefit from the text, the reader should construct programs in each of the paradigms, and write semantic specification, and implement a small programming language.

Organization

Since the subject area PL: Programming languages as described in the "ACM/IEEE Computing Curricula 199" consists of a minimum of 47 hrs of lecture, the text contains more material can be covered in a single course.

Pedagogy

The text pedagogical support through various exercise and laboratory projects. Some of the projects are suitable for small group assignment. The exercise include programming languages Concept for the students to gain maximum benefit from the text, the student should have access to a logic programming languages (such as Prolog), a modern functional language (such as Scheme, ML or Haskell), a concurrent programming languages (Ada, SR, or Ocaml), an object oriented programming languages (C++, Small-

Talk, Eiffel & Modula-3),
and a modern.

1.1 Models of Computation

Computation models begin
with a set of values.

The values can be
separated into two groups:
primitive and composite.

The primitive values (or type)
are usually numbers,
boolean values, and characters.

The composite values (or
type) are usually arrays,
records and recursively
defined values.

The Functional Model

The functional model of
computation consists of
a set of values,
functions, and the operation

of Function application
Function may be
named and may be
composed with other
functions.

The Logic Model

The logic Model of Comput
on is based on relat
and logic inference. Pro
grams consist of
definitions of relation
and computations are
inferences.

The imperative Model

The imperative model of
computation consists of
a state and the
operation of assignment
which is used to
modify the state.

Programs consist of sequences of commands and computations are change of the state.

Other Models

Programs in the concurrent programming model consist of multiple processes or tasks which may exchange information.

The computation may occur concurrently or in any order. Concurrency of programming is primarily concerned

Computability

The method of computation provided in a programming language. Most programming languages

utilize more than one
model of computation
implemented by the
programming language
One model of
predomination but I
provide imperative
constructs as well
While, Miranda.

1.2 Syntax and Semantics

The notice used in
the functional and logic
models tends to reflect
common mathematical
practice and thus,
it tends toward simplicity
and regularity. On
the other hand,
the notation used for
the imperative model

1.3

Pragmatics

Pragmatics is concerned about the usability of the languages, the application area, ease of implementation and use, and the language's success in fulfilling its design goals. For a language to have wide applicability it must make provision for abstraction, generalization and modularity. Abstraction permits the suppression of detail and provides constructs which permit the extension of programming languages.

1.4

Languages Design Principles

Programming languages are largely determined by the importance of the language designers to the area of readability, writeability and efficient execution. Some languages are largely determined by the necessity for efficient implementation and execution. Other are designed to be faithful to computational

1.5

Further Reading

For a programming languages which presents programming languages from the virtual machine point of view see part 2 [24]. For a programming languages from the virtual machine point of view of denotational semantics see Tennent [30]. For a programming languages which present programming languages from a programming methodology point of view see Hehner [11].