Gwani Software

Training Department

(Knowledge & Expertise)

Programming Language Design Curriculum

December 2011

Disclaimer

- Any trademark used belongs to its rightful owner.

Warning

Proposed by

Abubakar Muhammad

faqeer@gwanisoftware.com

Approved for usage by

Al-Ameen Abubakar, Director Training, this 5th day of Muharram, 1433 equivalent to

1/12/2011.

Gwani Software

TRAINING DEPARTMENT

Programming Language Design

General Description: - This course is intended to provide the basics concepts of designing a programming Language.

Aims: - The aims of this course are to:

1. Avail the trainee with knowledge about programming languages.
2. Avail the trainee with knowledge of translators.
3. Avail the trainee about classifications of programming languages.
4. Drill the trainee on designing of programming languages.

Objectives: - The trainee at the end of the training session should be able to;

- Know programming languages, their grammar, syntax, derivation tree, generations and common examples.
- Know about translators-Assemblers, Interpreters & Compilers.
- Know Imperative languages, ADT languages & OOP languages.
- Design a programming language.

Target Audience: - This course should be taken by Computer Scientists, Language designers, Compilers designers and anyone interested in language design.

Pre-requisite:- There is no any course standing as a pre-requisite to this course, however knowledge of any programming language is an added advantage.

Approximate Duration: - This course requires 5 weeks to complete.

Method of Assessment: - The trainee is to be assessed with a project work on design of a programming language.

Methodology: - The class takes a lesson discusses it and moves to the next lesson until all the lessons are adequately covered.

Recommended Resource Materials: - The following materials are considered resourceful to this course and are therefore recommended.

1. Prof. Alfred U. Aho, (2006), '**Trends in programming language design**'.
2. Ecma International, (2006), '**Eiffel: Analysis, Design & Programming Language**' Eana International.
3. Rapheal A. Finkel, (1996), '**Advanced Programming Language Design**'; University of Kentucky, Kentucky.
4. P. Sewell, J. J. Leifer, K. Wansbrough, F. Z. Nardelli, M. A. Williams, P. Harbouzit & V. Xafeiadis, '**Acute; High Level Programming Language Design for Distributed Computing**' University of Cambridge, Cambridge.
5. P. J. Landin, (1962), '**The next 700 Programming Languages Design'** Division of Sperry Rand Corp, New York.
6. M. Ordresky, P. Altherr, V. Crenet, I. Dragos, G. Dubochet, B. Emir, S. Medirmid, S. Micleloud, N. Mihaylov, M. Schmz, E. Stenman, L. Spoon, M. Zenger, (2006), '**An overview of the scalar programming language**' LAMP-REPORT.
7. Linda Melver & Domain Conway, ' **Seven deadly sins of introductory programming language design**' Monash University, Victoria.
8. Springer, G. & Fredman, D. P., (1992), '**Scheme & the art of Programming**' Massachusetts Institute of Technology, Massachusetts.

Week	Lesson
1.	**Introduction to programming Languages: -** Definitions, program elements, Grammar-Recursive grammar, formal language, syntax derivations tree. Generations: - 1st generation, 2nd generation, 3rd generation & 4th generation.
2.	**Translators: -** Definition, assembler, Interpreter, Compiler, Transition diagram, JVM.
3.	**Programming Languages types: -** Imperative languages, Abstract data type, Object oriented programming languages. OOP: - Encapsulation, Polymorphism, & Inheritance.
4.	**Language Design:-** consideration in programming language design abstraction, automation, defense in-depth, Manifest Interface, portability, regularity, simplicity, syntactic consistency, Information hiding, labeling, localized cost, orthogenality zero-one-infinity. Algorithm- Definition, Importance types & examples.

	Data- Definition, Importance types & examples. Structure- Definition, Importance types & examples. Application of the three- algorithm structure (structured programming construct), Data structure, Data Algorithm (1/0 assignment, logical operators), Data-structure-algorithm (program).
5.	**Review of languages: -** Review of Grammar, constructs and vocabulary of the following languages. QBASIC, C++, VBASIC, JAVA and COBOL.