



Gwani Software



TRAINING DEPARTMENT
(Knowledge & Expertise)

Programming Essentials Curriculum

December 2011

© Gwani Software Ltd, 2011.

All rights reserved.

Disclaimer

- Visual Basic is a trademark of Microsoft Corporation,
- And any other trademark belongs to its rightful owner.

Warning

This document is an exclusive property of Gwani Software Ltd, permission is granted to print and copy this document by trainees, instructors, supervisors and affiliated academies of Gwani Software likewise for non-commercial distribution by anyone.

With the exception of the above permission, no part of this document may be printed, copied, modified or used by anyone without a prior written permission is obtained from Gwani Software. Contravening this provision may lead to legal proceeding in a court of law.

Proposed by

Abubakar Muhammad

faqeer@gwanissoftware.com

Approved for usage by

Al-Ameen Abubakar, Director Training, this 9th day of Ramadan, 1433
equivalent to

28/7/2012.

Gwani Software

TRAINING DEPARTMENT

Windows XP

General Description: - This course is intended to provide basic technical skills needed in writing, testing, debugging and running computer programs.

Aims: - The aims of this course are to:

1. To give trainee technical knowledge on how best to write computer programs.
2. Avail the trainee with the common ways of testing computer programs.
3. Drill trainee on debugging of computer programs and running them efficiently.
4. Avail the trainee with the common features shared by most programming language.
5. Introduces the trainee to system calls as may be used in programming.
6. Drill the training on how to access external files from computer program.
7. Impact to the trainee the patience and discipline required in programming.

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

- Know how to write computer programs with ease
- Know how to test computer programs,
- Know how to debug computer programs.
- Know and use common features shared by most programming languages.
- Know how to use system calls in programming.
- Know how to access external files from a computer program.

- Know how to patiently follow programming task until the program is finally written.

Target Audience: - This course should be taken by programmers, computer scientist, software engineers, system designers, system analysts, instructors of programming and anyone interested in programming.

Pre-requisite:- Knowledge of Visual Basic, Java or C++ is a standing pre-requisite to this course. Knowledge of programming methodology is an added advantage.

Approximate Duration: - This course requires 21 hours of class session with practical.

Method of Assessment: - The trainee is to be assessed with practical assessment jobs to be given at a regular interval of time and examination at the end of the course.

Methodology: - The class takes a lesson discusses it and practices it and move to next. Programming assessment jobs are given to the trainee after series of lessons are covered. The class then moves to the next lesson in the same manner, until all the lessons are adequately covered.

Recommended resource materials: - The following are required for additional study regarding this course.

1. J. Bartlett, (2003), '**Programming from the ground up**'.
- 2.

Day 1	Introduction:- Definition of programming generation of programming language, machine language, assembly language, low level language, high level language, very high level language. Translators - interpreter & compiler.
Day 2	Introduction to Computer Architecture: - structure of computer memory, structure of CPU, program counter, instruction decoder, data bus, general-purpose registers, ALU.
Day 3	Programming: - a science and not art, flowchart design, flowchart design, pseudo code design, tracing of pseudo code.
Day 4	Programming Continued: - Factors to look upon on choice of programming

	language, system and application programming.
Day 5	System programming: Instructors code, design of assembly language program, assembling an assembly language program.
Day 6	Trainee's assessment: - the trainee should be given a programming job in assembly language.
Day 7	Application Programming: - Language vocabulary, syntax and semantics, lexis & structures, analysis of common programming languages not restricted but include Fortran, Basic, Java, C++, Cobol and Pascal.
Day 8	Functions: - definition, function name, function parameters – return value, passed value. Function declaration in selected programming languages not restricted but include Visual Basic, Java & C++. Calling function, in-built functions & user defined function. Function state – local, static and global. Recursive function.
Day 9	Trainees Assessment: - the trainee should be given a programming job on function design.
Day 10	File Processing: - Definition of external files, text files and non text files, file path, write operation, read operation, append operation, delete and update operation. Sample codes on file operations in common programming languages not restricted but include Visual Basic, Java and C++.
Day 11	Trainee's assessment: - the trainee should be given a programming job on file processing operations.
Day 12	Accessing database: - Database basics – field, records, files, unique key, primary key & knowledge bases. Common database not restricted but include MySQL, oracle and Ms Access. Discuss connection to databases in visual basic, Java & C++.
Day 13	Accessing database continued: - conduct insert, select, delete and update operations using any of Visual Basic, Java or C++.
Day 14	Trainee's assessment: - the trainee should be given a programming job on database connection and operations.
Day 15	Loop & controls: – Definition of loop & control, importance of loop in a program, infinite loop, importance of infinite loop in programming. Control structures in programming languages – if..else, break & continue, multi-selection. Analysis of loop & controls in visual basic, Java & C++.
Day 16	Program testing: – definition of program testing, importance of testing, unit test, integration test, test data. Factors to be considered during testing – relevance of test data, clean PC, documenting bug, fixing bugs, relevant user, concentrating on errors, period of testing, division of testing team from programming team, assigning error

	<p>codes, recovery points. Pre-release copies of programs, review & modifications.</p>
Day 17	<p>Optimizing programs: - definition of optimization, optimizing factors –speed, memory usage & disk space usage. Local optimization – pre-computing calculations, use of calculated results, locality of reference, register usage, inline functions, optimized instructions, data alignment. Global optimization – parallelization & statelessness.</p>
Day 18	<p>System calls: – Definition of system call, use in programming. Common Linux system calls not restricted but include exit, read, write, open, close, chdir, isseek, getpid, mkdir, rmdir, dup, pipe, brk, ioctl.</p>
Day 19	<p>Trainee’s assessment: – the trainee should be given a programming job on testing a program.</p>
Day 20	<p>Trainee’s assessment: – the trainee should be given a programming job on optimizing a program.</p>
Day 21	<p>Revision</p>