

Software Development Fundamentals (ITCA) + exam

Duration: 2 Days Course Code: ITCA-SDF

Overview:

This course introduces learners to key concepts and principles of software development. Candidates can gain insight into the key development methodologies, concepts of software development, standards, and software basics and learn through a hybrid of knowledge-based and hands-on training labs. **Continuing Professional Education (CPE): 16** **Exam voucher: 1 voucher included** (Valid for 12 months)

Target Audience:

A wide-range of individuals, including: Those new to IT, students, recent graduates and career changers. Audit, risk, security and governance professionals looking to gain base-line IT knowledge and skills. Current IT Professionals looking to reskill or upskill to broaden their IT knowledge and skills or keep up-to-date

Objectives:

■ 1- Software and its Evolution

- Identify and explain basic software concepts and terms
- Describe software types and their uses
- Summarize the history of software (mainframe, personal computers and software-as-a-service eras)

■ 2- Software Basics

- Describe algorithms (problem analysis, development, modeling, etc.)
- Differentiate between syntax, semantics, and lexis
- Differentiate between functional and object-oriented programming (OOP)

■ 3- Machine Language

- Explain Describe CISC and RISC instruction set architecture (ISA) families and their differences
- Identify machine-code concepts and language
- Define ASCII and Unicode, and interpreters and compilers

■ 4- High-level Languages

- Summarize scripting (e.g., Python, JavaScript, PowerShell and Bash)
- Recognize virtual machines (e.g., .NET, JAVA)
- Describe the portability of code
- Identify high-level languages (e.g., C, C++, Java, Python, HTML and .NET), procedural and nonprocedural languages, and graphical language

- Explain the flow of control structures (if/else)

■ 7- Repetition Constructs and Data Structures

- Explain the flow of control structures (while, do/while, etc.)
- Identify data structures

■ 8- Output

- Describe output types (peripherals, standard outputs [STDIO])
- Identify data storage types

■ 9- Software Development Standards

- Recognize ISO/IEC/IEEE standards (e.g., 12207:2017, 15289:2011, 730-2014, 90003)

■ 10- Software Development Life Cycle

- Describe SDLC management terms and concepts, including roles and responsibilities
- Identify the phases of software development life cycle
- Identify the phases of software development life cycle

■ 11- SDLC Models

- Describe software management methodologies (waterfall and Agile)
- Recognize Agile development methodologies (Scrum, XP, FDD, DevOps)
- Illustrate how to choose a software methodology
- Describe configuration management
- Explain software versioning and release best practices

- Describe an integrated development environment (IDE)

■ 5- Data Input

- Describe input types (standard input [STDIN], user input, process input, APIs)

- Recall data types (integers, characters, strings, etc.)

■ 6- Programming Basics

- Identify the basic structure of a program

- Compare synchronous vs. asynchronous processing

- Describe data processing

- Describe code repositories and proper code branching

■ 12- SDLC Security and Management

- Explain DevSecOps and Secure SDLC

- Identify project management tools and techniques

- Express software development ethics

Content:

1- Software and its Evolution

- What is Software?
- Software Types
- History of Software
- Mainframe Era
- Personal Computer and Desktop Software Era
- Cloud Computing Era
- Y2K Bug

2- Software Basics

- Algorithms
- Differences Among Syntax, Semantics, and Lexis
- Developing an Algorithm
- Analyze Problem
- Design Model
- Analyze Algorithm
- Develop Program
- Test Program
- Document Solution
- Functional vs. Object-Oriented Programming (OOP)
- Influential Software Enterprises

3- Machine Language

- Introduction
- Central Processing Unit
- Instruction Set
- CISC and RISC Instruction Set Architecture Differences
- Machine Code Language
- ASCII and Unicode
- Assembly Machine Language
- Interpreters and Compilers

4- High-level Languages

- Scripting
- Applications of Scripting Languages
- JAVA and .NET Virtual Machines
- Portability of Code
- Overview
- Procedural and Nonprocedural
- C and C++
- JAVA
- HTML
- Integrated Development Environment (IDE)
- Python
- .NET Suite and Visual C++

5- Data Input

- Input Data
- Process Input
- Data Types

6- Programming Basics

- Synchronous and Asynchronous Processing
- Boolean Logic and Operators
- Control Structures
- If Decision Making

7- Repetition Constructs and Data Structures

- While Loops
- Infinite While Loops
- Do While Loops
- For While Loops
- Nested Loops
- Data Structures

8- Output

- Output to Peripherals
- Data Storage Types
- Retrieving Information
- Updating Information
- Deleting Information

9- Software Development Standards

- Overview
- ISO/IEC/IEEE 12207:2017
- ISO/IEC/IEEE 15289:2011
- IEEE Std 730-2014
- ISO/IEC/IEEE 90003 (except SDLC)

10- Software Development Life Cycle

- ISO/IEC/IEEE 90003 (SDLC)
- Planning Phase
- Software Requirements Analysis
- Scheduling and Team Assembly
- Roles and Responsibilities
- Communication Plan
- Risk Management
- Benchmarking
- Designing Phase
- Implementation Phase
- Testing and Integration Phase
- Quality Assurance Responsibilities
- Quality Check Responsibilities
- Testing and Integration Automation
- Defect Analysis
- Deployment
- Maintenance Phase
- End of Life

11- SDLC Models

- SDLC Management
- SDLC Models—Waterfall
- SDLC Models—Agile
- The Agile Manifesto
- Agile Development Frameworks
- User Stories
- Assigning Effort Points
- Features and Epics
- Planning and Schedules
- Burndown and Velocity Charts
- Scrum
- SCRUM Roles and Responsibilities
- Extreme Programming (XP)
- Feature-Driven Development (FDD)
- Choosing a Methodology
- Versioning and Repositories
- Software Versioning Schemes
- Repository Mechanics
- Repository Code Branches Structure

12- SDLC Security and Management

- DevSecOps
- Difference Between DevOps and DevSecOps
- Secure SDLC
- Project Management Tools and Techniques
- Time Management and Scheduling Techniques
- Computer Software Development Ethics

Practice Labs

- Number Systems
- Debugging
- Command Creation
- Scripting with Python
- Integrated Development Environments (IDEs)
- GCC and C
- Batch and Bash Scripting
- Standard Input and Output

Further Information:

For More information, or to book your course, please call us on 00 20 (0) 2 2269 1982 or 16142

training@globalknowledge.com.eg

www.globalknowledge.com/en-eg/

Global Knowledge, 16 Moustafa Refaat St. Block 1137, Sheraton Buildings, Heliopolis, Cairo