B001 : Introduction to Software Development Methodologies

Overview:

Software development is the set of activities and processes that will eventually result in a software product. This may include inventing, improving, selecting among alternative solutions, and then describing computer programs that meet user requirements within the constraints of the environment. Adopting a good software development methodology will eventually lead to a software product that meets most, if not all of the agreed upon requirements, within the given constraints that, which includes cost, and schedule.

Objective:

The objective of this introductory course is to provide participants with the basics of common software development methodologies with emphasis on concepts and techniques of software development that will form the base for subsequent courses.

Topics:

- Introduction to Systems Engineering.
- Basics of Software Engineering.
- Software Process Models
- The Software Development Life Cycle
- Software Development Tools
- Managing Software Development Projects

Duration:

3 Days

Pre-Requisites:

None

B002: **Software Requirements**

- **Objectives:**

  The main objective of this course is to provide trainees with the basics of software requirements engineering including requirements elicitation techniques, analysis models and methodologies, tools and management issues.

- **Outcomes:**

  Upon successful completion of this course, trainees will be able to:

  - Understand the types of software requirements.
  - Understand requirements various elicitation techniques
  - Develop & document software requirements at different levels following standard techniques.
  - Develop software analysis models (using UML standard notations)
  - Understand issues related to software requirements validation and management

- **Contents:**

  - Introduction
  - Characteristics of Good Requirements
  - Types of requirements
  - Requirement engineering process
  - Developing and Documenting Software Requirements
  - Documentation of software requirements
  - Software requirement validation
  - Requirements Management
  - Software requirements tools.

- **Duration:**

  2 Days

- **Pre-Requisites:**

  Introduction to Software Development Methodologies.

B003 : Process Modeling

- Objectives :

  The goal for this course is to give the trainee a clear picture on function-based process modeling. This includes a discussion of what process modeling is, the goals of process modeling, and a comparison of the different methods used for model building with emphasis on graphical (visual) representation. Detailed information on how to collect requirements and to construct appropriate models should be covered. Case studies that illustrate examples may be presented and examples solved as exercises using graphical notation.

- Outcomes :
  The participants will receive detailed training on modeling a business process. They will learn not only the technical aspects of the covered techniques, but also the methodology they should follow in using them.
  Business analysts, designers or modelers will gain a set of core techniques that they can apply regardless of the tool or notation chosen.
  The course lays the foundation for building business process models that are understandable and shareable so that one can be confident that he/she will make the right modeling choices. When reviewing someone else's models, it will help them understand and ask the right questions.

- Contents :
  - Introduction to Process Modeling
  - Business Process Management (BPM)
  - Requirements Collection for Process Modeling
  - Business Modeling Languages (e.g. UML, BPEL, BPMN, WS-CDL)
  - Defining Process Models
  - Business Process Modeling Techniques
  - Data Flow Diagrams
  - Work Flows
  - UML Activity Diagrams
  - Measuring and Analyzing Processes

- Duration :
  3 Days

- **Pre-Requisites:**
  Introduction to Software Development Methodologies.

B004 : Data Modeling

- Objectives :

  As it is becoming more essential for every IT professional to have a basic understanding of data modeling, this course aims to provide an opportunity for the trainees to gain fundamental data modeling skills that all developers are expected to possess. By getting exposed to a number of examples and involved in the creation of data models, through a series of exercises, the trainees will be able to read an existing data model, create a new data model, and appreciate fundamental database design techniques.

- Outcomes :

  The trainees will gain an understanding of relational database concepts and database implementation issues. Upon completion of the course, the trainees are expected to be well acquainted with entity relationship (ER) data modeling and data normalization, the creation and management of conceptual data models, generation of physical data models, databases and implementation details.

- Contents :

  - Introduction and Basic Concept
  - Overview of Data Modeling Using The Entity Relationship Diagram
  - Relationships
  - Relationship Degrees
  - Complex Entities
  - ERD Design Guidelines and Quality Checks
  - Object Modeling and Class Diagrams
  - Mapping Entity Relationship Diagram Into Tables
  - Data Normalization
  - Designing Databases

- Duration :
  3 Days

- **Pre-Requisites:**

  Introduction to Software Development Methodologies.

B005 : Software Design I

- **Objectives:**

The main objective of this course is to provide trainees with the basics of software design including architectural design and fundamental activities of low level design.

- **Outcomes:**

Upon successful completion of this course, trainees will be able to:

- Understand the context of software design within the software development process
- Master quality characteristics of software design
- Develop & document software detailed design using UML 2.0.
- Understand the process of Object Oriented Software Design
- Develop & document software architecture following standard techniques.
- Master the principals of designing user interfaces.

- **Contents:**

- Introduction
- Software Design in Context
- Characteristics of Good Design
- Design Modeling using UML
- Object Oriented Design
  - Overview
  - From Analysis To Design
  - Assigning Responsibilities Using Interaction Diagrams
  - Completing the Design Class Diagrams
- Software Architecture Design
  - Introduction
  - Developing Software Architecture
- Designing User Interface Layers

- **Duration:**

3 Days

- **Pre-Requisites:**

Introduction to Software Development Methodologies.

B006 : Software Design II

- **Objectives:**
  The main objective of this course is to provide trainees with the basics of software design and architecture technologies such as Design Patterns, components design, Software Product Lines, Aspect Oriented Architecture, and Model-Driven Architecture. The course also aims at giving hands on experience on tools supporting software design.

- **Outcomes:**
  Upon successful completion of this course, trainees will be able to:
  - Develop software detailed designs using design patterns
  - Develop software using state of the art Service Oriented Architecture Technologies
  - Understanding emerging architectural models; e.g.
    - Software Product Lines
    - Aspect Oriented Architecture
  - Understand new trends in software architecture development
    - Model-Driven Architecture
    - Middleware for distributed systems
  - Develop high quality software design document
  - Develop selection criteria for software design automated tools

- **Contents:**
  - Introduction
  - Review of Design Fundamentals
  - Patterns
  - Specifying Component Interfaces
  - Aspect Oriented Architecture
  - Model Driven Architecture
  - Software Product Lines
  - Middleware for Distributed Systems
  - Service Oriented Architecture and Technologies
  - Designing the Data Access Layers
  - Documenting Software Design
  - Automated Tools for Software Design

- **Duration:**
  3 Days

Pre-Requisites:

B005 : Software Design I

B007 : Software Quality Assurance and Testing

Overview:
Software Quality is the successful satisfaction of explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software. Software testing on the other hand is the process and activities associated with confirming the proper operation of a software system according to the agreed upon requirements.

Objective:
The main objectives of this course are to provide participants with the basic concepts, techniques, tools, and methodologies for software Quality assurance and testing.

Topics:

- Introduction to Software Quality
- Software Dependability
- Software Quality Assurance and Testing
- Software Quality Metrics
- Standards in Software Quality
- Software Quality Assurance Plan and Activities
- Software Reviews
- Verification and Validation
- Software Testing
- Test Documentation (Plan, Procedure, etc.)
- Checklists and Operational Profiles
- Different Types of Tests
- Introduction to Models for Software Testing
- Beyond Testing: Process Improvement

Duration:
3 Days

Pre-Requisites:

Introduction to Software Development Methodologies.