

**Course Description:** This course gives an overview of the object-oriented approach to software development with emphasis on the responsibilities and expectations of the Business Analyst. It introduces the concepts of OO development and emphasizes the usage of the Unified Modeling Language (UML) to develop and document software design. Special emphasis is placed on those aspects of OO development where Business Analysts are involved.

**Who Should Attend:** This course is specifically for Business Analysts, who want to gain an object-oriented perspective to the software development life-cycle. It will also be beneficial to Project Managers, System Analysts, and Developers seeking the same knowledge.

**Prerequisites:** The student should have a general knowledge of the role of the Business Analyst — gained through experience or taking courses that precede this course in this curriculum path.

**Benefits of Attendance:** Upon completion of this course, students will be able to:

- Discuss the fundamentals of developing software using object-oriented technology.
- Describe the process of end-to-end software development using the principles of object-orientation.
- Use the Unified Modeling Language to define early portions of a software system.
- Demonstrate the place of use cases in the process of requirements management.
- Incorporate the standard principles of requirements elicitation, and record and change management into an object-oriented project.
- Associate the demands and importance of software testing with the principles of object-oriented development.
- Use the UML diagrams with a business analysis perspective.

### Course Outline:

#### Introduction to Object-Oriented Development

What is object-oriented software development?  
 Typical software life cycle  
 Differences in OO and Procedural software life cycle  
 Characteristics of the OO development process  
 Benefits of using the OO development process

#### Principles of Object-Oriented Technology

What is an object?  
 What is a class?  
 Relationship of classes to objects  
 Three major features of object-orientation  
 Packages in OO development  
 How classes, objects, and packages relate to business analysis tasks

#### Visual Modeling and the UML

Why use models in software development?  
 What is the Unified Modeling Language?  
 UML diagrams useful for the BA  
 Other useful non-OO diagrams for visual modeling  
 Business Process Models  
 Data Flow Diagrams  
 Context Diagrams  
 The OO development process reprised with UML diagrams

#### Developing Business Use Cases

Identifying major system features  
 Discovering stakeholders  
 Eliciting requirements and features from stakeholders  
 More on use case diagrams  
 Use case specifications expanded  
 Requirements not covered with business use cases  
 Evolving business use cases into system use cases

#### Activity Diagrams

The versatility of activity diagrams  
 Associating activity diagrams with business and system use cases

#### Capturing Static System Definition with Class Diagrams

What is a class diagram?  
 Features of class diagramming  
 Class associations  
 Association  
 Aggregation  
 Inheritance  
 Application of class diagramming to business analysis tasks

#### Other Useful UML Diagrams for the Business Analyst

Interaction diagrams and their use  
 Capturing system dynamics with sequence diagrams  
 Machine diagrams and their use  
 Capturing static system features with state machine diagrams  
 Package diagrams and their use

Capturing system organization with package diagrams

#### Applying OO Principles of Reuse and Encapsulation to BA Tasks

Two major principles of the OO methodology - reuse and encapsulation  
 Abstracting the OO principle of encapsulation and applying it to the BA  
 Separation of responsibility, information, and dynamics in BA processes  
 Examples of BA process encapsulation  
 Exploiting the OO principle of reuse in business analysis  
 Examples of reuse in business analysis activities

#### Object Oriented Testing and the BA

What is object-oriented testing?  
 Applying OO principles to general software testing  
 Testing OO systems

#### Summary

Putting it all together  
 What happens to the requirements when they leave the BA?  
 Review the OO process from the BA perspective