

## COURSE PROFILE: A BASIC UNDERSTANDING OF OBJECT ORIENTED PROGRAMMING

<b>Title</b>	A Basic Understanding of Object Oriented Programming
<b>Length</b>	5 days
<b>Description</b>	This introductory course takes a language-independent, object-oriented approach to programming design and logic. It provides a solid foundation needed to understand how object oriented business applications and Web sites are developed. This course introduces classes, objects, behaviors, attributes, and other object-oriented topics such as inheritance and exception handling.
<b>Target Audience</b>	This course targets all IT staff members (systems analysts, managers, and other non-programming positions) who are involved in planning, designing, and developing object oriented applications and Web sites.
<b>Prerequisites</b>	None. IT Basics is recommended.
<b>Learning Objectives</b>	<p>The student will be able to describe:</p> <ul style="list-style-type: none"><li>• the evolution of programming techniques and processes,</li><li>• object-oriented programming concepts,</li><li>• using methods and parameters,</li><li>• program structure,</li><li>• decisions constructs,</li><li>• looping,</li><li>• using arrays,</li><li>• event-driven programming with graphical user interfaces (GUI),</li><li>• modeling with UML,</li><li>• polymorphism</li><li>• inheritance, and</li><li>• exception handling.</li></ul>
<b>Course Material</b>	<p>Each Student will receive:</p> <ul style="list-style-type: none"><li>• a college level text book on the principles of Web design and</li><li>• slide material.</li></ul>
<b>Cost</b>	<p>Our all inclusive local onsite fee for up to 15 students is \$12,650.00*.</p> <p>*Effective Date: 01/01/2008. This price does not include travel outside the Baltimore, MD or District of Columbia area. This does not include any offsite facility charge.</p>

## COURSE PROFILE: A BASIC UNDERSTANDING OF OBJECT ORIENTED PROGRAMMING

### Course Content

#### Lesson 1 - An Overview of Computers and Logic

- Evolution of programming techniques
- Steps involved in the programming process
- Flowcharts and pseudocode statements
- Application class with a **main()** method
- Variables and data types

#### Exercise

Identify Objects in the case study.

#### Lesson 2 - Object-Oriented Programming Concepts

- Basic principles of OOP
- Defining classes and creating class diagrams
- Understanding public and private access
- Instantiating and using objects
- Inheritance
- Polymorphism
- Describe GUI classes as an example of built-in classes

#### Exercise

Develop a class diagram for the case study.

#### Lesson 3 - Using Methods and Parameters

- Methods with and without arguments
- Instance methods in a class
- Rationale behind data hiding
- Organize classes
- Understand the role of the “**this**” reference
- Constructors

#### Exercise

Develop pseudocode for one of the case study classes.

#### Lesson 4 - Understanding Structure

- Structure in object-oriented methods
- Flowcharting
- Three basic structures of sequence, selection, and loop
- The need for structure
- Recognizing structure
- Two special structure: case and do until

#### Exercise

Draw a flow chart for the case study

## COURSE PROFILE: A BASIC UNDERSTANDING OF OBJECT ORIENTED PROGRAMMING

### **Course Content, continued**

#### **Lesson 5 - Making Decisions**

- Boolean expressions used to make comparisons
- Relational comparison operators
- AND logic
- OR logic
- Selections within ranges
- Common errors using range checks
- Precedence when combining AND and OR selections
- **Case** structure
- Decision table

#### **Exercise**

Develop decisional pseudocode for one of the case study classes.

#### **Lesson 6 - Looping**

- Advantages of looping
- Control loops with variables, counters, and sentinel values
- Common loop mistakes
- For loop
- Do until loop
- Characteristics shared by all loops
- Nesting loops
- A loop to accumulate totals

#### **Lesson 7 - Arrays**

- Arrays and how they occupy computer memory
- Loading array values from a file
- Searching an array for a match
- Using a constant to store an array size

#### **Lesson 8 - Advanced Array Concepts**

- Remaining within array bounds
- Using a **for** loop to process arrays
- Declaring an array of objects
- Passing arrays to methods
- Sorting array elements
- Sorting arrays of objects
- Two-dimensional and multi-dimensional arrays
- Built-in **Arrays** class

## COURSE PROFILE: A BASIC UNDERSTANDING OF OBJECT ORIENTED PROGRAMMING

### **Course Content, continued**

#### **Lesson 9 - Event-Driven Programming with Graphical User Interfaces**

- Principles of event-driven programming
- Actions that GUI components can initiate
- Designing graphical user interfaces
- Modifying the attributes of GUI components
- Steps to developing an event-driven application
- Multithreading
- Animation
- Margin, padding, border, and special box properties

#### **Exercise**

Design a customer data entry screen.

#### **Lesson 10 - Modeling with UML**

- UML and its need
- Use case diagrams
- Sequence and collaboration diagrams
- Statechart diagrams
- Activity diagrams
- Component and deployment diagrams

#### **Exercise**

Create a partial use case diagram for the case study.

#### **Lesson 11 Object Concepts: Polymorphism and Inheritance**

- Blocks and scope
- Overload methods and constructors without ambiguity
- The **this** reference
- Using inheritance to extend classes
- Overriding superclass methods
- Using superclass constructors
- Information hiding
- How inheritance promotes good software design

#### **Lesson 12 Exception Handling**

- About exceptions
- The limitations of traditional error handling
- Try and catch exceptions
- Using the finally block
- Advantages of exception handling
- Exceptions a method can throw