

IT3105: Object Oriented Analysis and Design (Compulsory)

INTRODUCTION

This course starts with object oriented concepts and moves towards the preparation of standard UML diagrams using an UML modeling tool.

CREDITS: 03

LEARNING OUTCOMES

After successfully completing this course you will be able to:

- Describe Object Oriented Analysis and Design concepts and apply them to solve problems
- Prepare Object Oriented Analysis and Design documents for a given problem using Unified Modeling Language

MINOR MODIFICATIONS

When minor modifications are made to this syllabus, those will be reflected in the Virtual Learning Environment (VLE) and the latest version can be downloaded from the relevant course page of VLE. Please inform your suggestions and comments through the VLE. <http://vle.bit.lk>

ONLINE LEARNING MATERIALS AND ACTIVITIES

You can access all learning materials and this syllabus in the VLE: <http://vle.bit.lk>, if you are a registered student of BIT degree program. It is very important to participate in learning activities given in the VLE to learn this subject.

ONLINE ASSIGNMENTS

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). Maximum mark for a question is 10, minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering 40% of assignment quiz 1 and 60% of assignment quiz 2. Pass mark for the online assignments in a course is 50. You are advised to do online assignments before the final exam of the course. It is compulsory to pass all online assignments to partially qualify to obtain year 2 certificate.

FINAL EXAMINATION

Final exam of the course will be held at the end of the semester. Each course in the semester 3 is evaluated using a two hour question paper which consists of 20-25 MCQs and 3-4 structured questions based on a given case study.

OUTLINE OF THE SYLLABUS

Topic	Minimum number of hours
1. Introduction to Object Oriented Concepts	02
2. Object Oriented Analysis and Modeling	01
3. Software Development Process	02
4. Creating Use Case Diagrams	05*
5. Identifying Classes ,Packages and drawing Class diagrams, Object Diagrams	06*
6. Object Oriented Design and Modeling using UML	04*
7. Working with State diagrams	03*
8. Discovering Object Interactions	05*
9. Working with Activity Diagrams	03*
10. Component and Deployment Diagrams	01
11. New diagrams in UML 2.x , Model Driven Architecture (MDA), Executable UML	03
12. Case Studies	10*
Total hours	45

* Students are expected to do practicals and tutorials to strengthen their knowledge of these sections.

REQUIRED MATERIALS

Main Reading

Ref1: Systems Analysis and Design Methods by Jeffrey L. Whitten, Lonnie D. Bentley, 7th edition, ISBN 0-07-058224-6, Tata McGraw-Hill, 2007.

Ref2: Teach Yourself UML in 24 Hours, Joseph Schmuller, 3rd Edition, ISBN 81-297-0609-1, Pearson Education, 2004

Ref3: Software Engineering, Ian Sommerville, 9th edition, ISBN 978-81-317-6216-5, Pearson, 2011,

Ref4: http://en.wikipedia.org/wiki/Rational_Unified_Process

Ref5: <http://www.uml-diagrams.org>

Supplementary Reading

Ref6: UML 2 Bible, Tom Pender, Wiley Publishing, Wiley Publishing Inc, 2003

Ref7: “UML User Guide”, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 2002.

Ref8: Visual Modeling with Rational Rose 2002 and UML By Terry Quatrani Foreword by Grady Booch, 2002

Ref9: The Rational Unified Process: An Introduction, Philippe Kruchten, 3rd edition, Addison-Wesley Professional, 2003

DETAILED SYLLABUS

1. Introduction to Object Oriented Concepts (02 hrs.)

Learning Objectives

- Identify and describe the Object Oriented concepts

Material /Sub Topics

1.1 Classes, Objects [Ref2: pg. 32, 33]

1.2 Abstraction [Ref2: pg. 34]

1.3 Inheritance [Ref2: pg. 35]

1.4 Polymorphism [Ref2: pg. 36]

1.5 Encapsulation [Ref2: pg. 37]

1.6 Message Sending [Ref2: pg. 38]

1.7 Associations [Ref2: pg. 40]

1.8 Aggregation and Composition [Ref2: pg. 41-43]

2. Object Oriented Analysis and Modeling (01 hrs.)

Learning Objectives

- Define object modeling and explain its benefits
- Define Unified Modeling Language (UML) and its various types of diagrams

Material /Sub Topics

2.1 Introduction to Object Oriented Analysis [Ref2: pg. 370-371][Ref3]

2.2 Introduction to Unified Modeling Language (UML) [Ref2: pg. 7-27][Ref5]

3. Software Development Process (02 hrs.)

Learning Objectives

- Recognize the benefits of a software development process
- Identify the different Process Models available
- Describe the stages of Rational Unified Process (RUP)

Material /Sub Topics

3.1 Introduction [Ref3: pg. 28-30]

3.2 Process Models [Ref3: pg. 29-36]

3.2.1 Waterfall Model

3.2.2 Spiral Model [Ref3:pg 49]

3.2.3 Incremental Development

3.2.4 Reuse-Oriented Software Engineering

3.3 Introduction to Rational Unified Process [Ref3: pg. 50-53] [Ref: 4]

3.4 Introduction to Agile Software Development [Ref3: pg. 57-62]

4. Creating Use Case Diagrams (05 hrs.)

Learning Objectives

- Describe the benefits of Use-Case Modeling
- Define actors, use cases and use-case relationships
- Identify and describe the steps for preparing a use-case model

Material /Sub Topics

4.1 An Introduction to Use-Case Modeling [Ref1: pg. 244-250] [Ref: 5]

4.1.1 Actors

4.1.2 Use Cases

- 4.1.3 Use Case Relationships
- 4.2 The Process of Requirements Use-Case Modeling [Ref1: pg. 251-262]
 - 4.2.1 Identify Business Actors
 - 4.2.2 Identify Business Requirements Use Cases
 - 4.2.3 Construct Use-Case Model Diagram
 - 4.2.4 Document Business requirements Use-Case Narratives
 - 4.2.5 Use Cases and Project Management
- 4.3 The Process of Object Modeling [Ref1: pg. 383-390]
 - 4.3.1 Introduction
 - 4.3.2 Constructing the Analysis Use-Case Model

5. Identifying Classes, Packages and drawing Class Diagrams, Object Diagrams (06 hrs.)

Learning Objectives

- Recognize the concepts and notations used for finding Objects and Classes
- Identify the UML concepts of Stereotypes and Packages
- Draw Package Diagrams, Class Diagrams, Object Diagrams
- Illustrate the definitions of relationships between classes in the system. Specifically, the concepts of Association and Aggregation
- Define Reflexive and Package relationships
- Illustrate the application of generalization and specialization principles to discover super class/subclass relationships

Material /Sub Topics

- 5.1 Visualizing a Class [Ref 2: pg. 47-56]
- 5.2 Working with Relationships [Ref 2: pg. 61-87, 225-229]
 - 5.2.1 Associations
 - 5.2.2 Inheritance and Generalization
 - 5.2.3 Class Diagrams and Object Diagrams
 - 5.2.4 Package Diagrams
 - 5.2.5 Understanding Aggregations, Compositions
 - 5.2.6 Interfaces and Realizations

6. Object Oriented Design and Modeling using UML (04 hrs.)

Learning Objectives

- Differentiate between the different types of object classes
- Identify the design relationships

- Define visibility and explain its three levels
- Recognize the concept of object responsibility and how it is related to message sending between object types
- Describe the activities involved in object-oriented design
- Differentiate between a design use-case narrative and an analysis use-case narrative
- Describe the object reusability and the use of design patterns

Material /Sub Topics

6.1 Design of an Object Oriented System [Ref1: pg. 647-651]

- 6.1.1 Introduction
- 6.1.2 Different types of Object Classes
- 6.1.3 Design Relationships
- 6.1.4 Attribute and Method Visibility
- 6.1.5 Object Responsibilities

6.2 The Process of Object Oriented Design [Ref1: pg. 651-666]

- 6.2.1 Refining the Use Case Model
- 6.2.2 Modeling Class Interactions, Behaviours, and States that support use case scenario
- 6.2.3 Updating the Object Model to Reflect the Implementation Environment

6.3 Object Reusability and Design Patterns [Ref1: pg. 666-671]

7. Working with State Diagrams (03 hrs.)**Learning Objectives**

- Draw state transition diagrams that provide additional analysis techniques for classes with significant dynamic behaviour

Material /Sub Topics

7.1 An introduction to State Diagrams [Ref2: pg. 123-130][Ref5]

- 7.1.1 Adding details to States
- 7.1.2 Adding details to Transitions
 - 7.1.2.1 Events
 - 7.1.2.2 Actions
 - 7.1.2.3 Guard Conditions

7.1.3 Substates

7.2 UML 2.x State Diagrams [Ref2: pg. 130-131] [Ref5]

7.3 Importance of State Diagrams [Ref2: pg. 131-132]

8. Discovering Object Interactions (05 hrs.)

Learning Objectives

- Add scenarios to the system to describe how Use Cases are realized as interactions among societies of objects
- Describe a scenario by applying sequence diagrams and communication diagrams

Material /Sub Topics

8.1 Sequence Diagrams [Ref2: pg. 135-153] [Ref5]

8.1.1 Introduction

8.1.2 Framing a Sequence Diagram in UML 2.x

8.1.3 Combined Fragment

8.2 Communication Diagrams [Ref2: pg. 157-168]

9. Working with Activity Diagrams (03 hrs.)

Learning Objectives

- Model Use-Case activities using Activity Diagrams
- Identify and draw swimlanes in activity diagrams

Material /Sub Topics

9.1 Modeling the Use-Case activities using Activity Diagrams [Ref1: pg 390-394, Ref2: pg. 173-181]

9.2 New Concepts from UML 2.0 [Ref2: pg. 181-191]

10. Component and Deployment Diagrams (01 hr.)

Learning Objectives

- Describe Component diagrams and Deployment diagrams that are used to graphically show the physical architecture of the software and hardware of the system

Material /Sub Topics

10.1 Component Diagrams [Ref2: pg. 197-209, Ref1: pg. 671-674]

10.2 Deployment Diagram [Ref2: pg. 213-222, Ref1: pg. 671-674]

11. New Diagrams in UML 2.x , Model Driven Architecture (MDA) , Executable UML(03 hrs.)**Learning Objectives**

- Describe the Composite Structure Diagram that shows the internal structure of classes
- Describe the Interaction Overview Diagram that expands the activity diagram
- Describe Timing Diagram which is designed to show long an object is in a state
- Describe Profile Diagram which allows to define custom stereotypes, tagged values and constraints

- Describe Model Driven Architecture and Executable UML

Material /Sub Topics

11.1 New Diagrams in UML 2.x

11.1.1 Composite Structure Diagram [Ref2: pg. 22]

11.1.2 Interaction Overview Diagram [Ref2: pg. 23]

11.1.3 Timing Diagram [Ref2: pg. 24]

11.1.4 Profile Diagram [Ref5]

11.2. Model Driven Architecture and Executable UML [Ref 3: pg. 138-142][Ref 4]

12. Case Studies (10 hrs.)**Learning Objective**

- Draw standard UML diagrams using an UML modeling tool for a given case study

Hardware and Software Requirements**Hardware**

Any standard PC (Pentium)

Software

- Windows 95/98/2000/NT/XP/Vista
- ArgoUML OR Microsoft Visio OR Rational Rose 2002 OR StarUML OR any tool that supports UML 1.4 and higher