

Program Content

Semester	III	
Course Code:	IT 3106	
Course Name:	Object Oriented Analysis and Design	
Credit Value:	3	
Core/Optional	Core	
Hourly Breakdown	Theory	Independent Learning
	45 hrs	105 hrs
<p>Course Aim: To provide necessary skills and competencies to analyze and design a system using object-oriented approach</p> <p>Intended Learning Outcomes: After following this course, students should be able to</p> <ul style="list-style-type: none"> - describe Object Oriented Analysis and Design concepts and apply them to solve problems - define UML (Unified Modeling Language) and its various types of diagrams - prepare Object Oriented Analysis and Design documents for a given problem using Unified Modeling Language - explain the transition from analysis to design - describe the key activities in the component-based software engineering (CBSE) process <p>Required Tools</p> <p>ArgoUML OR StarUML OR Visual Paradigm-Community Edition OR any tool that supports UML 1.4 and higher</p>		
Course Content: (Main Topics, Sub topics)		
Topic	Theory (Hrs)	
1. System Concepts for Object Modelling	03	
2. Object Oriented Software Development Process	03	
3. Computer Aided Systems Engineering	02	
4. Business Process Identification with Use Case Modelling	05*	
5. Business Process Modelling with Activity Diagrams	03*	
6. Structural Modelling using Class, Package Diagrams	06*	
7. Behavioural Modelling using Interaction Diagrams	05*	
8. Behavioural State Machines and Other UML diagrams	05*	
9. Object Oriented Design	03	
10. Component Based Software Engineering	03	
11. System Modelling Using Case Studies	07*	
	Total	45
* Students are expected to do practical's and tutorials to strengthen their knowledge of these sections.		

1. System Concepts for Object Modelling (3 hours)

- 1.1. Classes and Objects [Ref 1: Pg. 19-20]
- 1.2 Methods and Messages [Ref 1: Pg. 20]
- 1.3 Encapsulation and Information Hiding [Ref 1: Pg. 20]
- 1.4 Inheritance [Ref 1: Pg. 21]
- 1.5 Polymorphism and Dynamic Binding [Ref 1: Pg. 22]
- 1.6 Introduction to Unified Modeling Language (UML) [Ref 1: Pg. 34-35, Ref 4]

2. Object Oriented Software Development Process (3 hours)

- 2.1 Introduction to Object Oriented Systems Analysis and Design [Ref 1: Pg. 23-25]
- 2.2 Benefits of OOAD [Ref 1: Pg. 25]
- 2.3 The Unified Process [Ref 1: Pg. 25-34]
 - 2.3.1 Phases
 - 2.3.2 Workflows

3. Computer Aided Systems Engineering (2 hours)

- 3.1 Introduction to CASE and CASE tools [Ref 1 Pg 77]
- 3.2 Evolution of CASE tools [Ref 1: 77-79 , Ref 3, Ref Teacher's Note]
- 3.3 Popular Classification of CASE tools [Ref 1 77, 79 ,Ref 3, Ref Teacher's Note]

4. Business Process Identification with Use Case Modelling (5 hours)

- 4.1 Introduction to Use-Case Modeling [Ref 1: Pg. 119-121]
- 4.2 Elements of a Use Case Diagram [Ref 1: Pg. 121-126]
 - 4.2.1 Actors
 - 4.2.2 Use Cases
 - 4.2.3 Use Case Relationships
- 4.3 Creating a Use Case Diagram [Ref 1: Pg. 126-129]
- 4.4 Business Process Documentation with Use Cases and Use-Case Descriptions [Ref 1: Pg. 140-152]
 - 4.4.1 Elements of a Use-Case Description
 - 4.4.2. Creating Use-case Descriptions

5. Business Process Modelling with Activity Diagram (3 hours)

- 5.1 Introduction to Activity Diagrams [Ref 1: Pg. 129-130]
- 5.2 Elements of an Activity Diagram [Ref 1: Pg. 131-136]
 - 5.2.1 Actions and Activities
 - 5.2.2 Object Nodes
 - 5.2.3 Control Flows and Object Flows
 - 5.2.4 Control Nodes (Initial, final-activity, final-flow, decision, merge, fork, and join)
- 5.1.4 Swimlanes
- 5.3 Creating Activity Diagrams [Ref 1: Pg. 136-140]
- 5.4 Verifying and Validating the Business Processes and Functional Models [Ref 1: Pg. 153-157]

6. Structural Modelling using Class Diagrams (6 hours)

- 6.1 Introduction to structural modeling [Ref 1: Pg. 163-164]
- 6.2 Basic elements of structural models [Ref 1: Pg. 164-166, Ref 4]
 - 6.2.1 Classes, Attributes, and Operations
 - 6.2.2 Relationships

- 6.2.2.1 Association
- 6.2.2.2 Aggregation, Composition
- 6.2.2.3 Generalization

6.3 Object Identification [Ref 1: Pg. 166-172]

- 6.3.1 Textual Analysis
- 6.3.2 Brainstorming
- 6.3.3 Common Object Lists
- 6.3.4 Patterns

6.4 CRC Cards [Ref 1: Pg. 172-175]

- 6.4.1 Responsibilities and Collaborations
- 6.4.2 Elements of a CRC Card
- 6.4.3 Role-Playing CRC Cards with Use Cases

6.5 Class Diagrams [Ref 1: Pg. 176-184]

6.6 Creating Structural Models using CRC cards and Class Diagrams [Ref 1: Pg. 185-194]

6.7 Verifying and Validating the Structural Model [Ref 1: Pg. 194-197]

7. Behavioural Modelling using Interaction Diagrams (5 hours)

7.1 Introduction to Behavioural Modeling using Interaction Diagrams [Ref 1: Pg. 203-204]

7.2 Sequence Diagrams [Ref 1: Pg. 204-215]

- 7.2.1 Elements of a Sequence Diagram
- 7.2.2 Creating Sequence Diagrams

7.3 Communication Diagrams [Ref 1: Pg. 216-221]

- 7.3.1 Elements of a Communication Diagram
- 7.3.2 Creating a Communication Diagram

8. Behavioural State Machines and Other UML diagrams (5 hours)

8.1 Behavioural State Machines [Ref 1: Pg. 221-229]

- 8.1.1 States, Events, Transitions, Actions, and Activities
- 8.1.2 Elements of a Behavioral State Machine
- 8.1.3 Guidelines for Creating Behavioral State Machines
- 8.1.4 Creating a Behavioral State Machine

8.2 CRUDE Analysis [Ref 1: Pg. 229-232]

8.3 Verifying and Validating the Behavioral Model [Ref 1: Pg. 233-235]

8.4 Other UML Diagrams [Ref 1: Pg. 34-36, Ref 4]

Timing, interaction Overview, Composite Structure, Component, Deployment and Profile Diagrams

9. Object Oriented Design (3 hours)

9.1 Introduction to Object Oriented Design [Ref 1: Pg. 240-241]

9.2 Verifying and Validating the Analysis Models [Ref 1: Pg.242-257]

- 9.2.1 Balancing Functional and Structural Models
- 9.2.2 Balancing Functional and Behavioral Models
- 9.2.3 Balancing Structural and Behavioral Models

9.3 Evolving Analysis Models into Design Models [Ref 1: Pg. 257-262]

9.4 Package Diagrams [Ref 1: Pg. 262-268]

- 9.4.1 Guidelines for Creating Package Diagrams
- 9.4.2 Creating Package Diagrams
- 9.4.3 Verifying and Validating Package Diagrams

10. Component Based Software Engineering (3 hours)

10.1 Introduction to CBSE [Ref2: 465-467]

10.2 Components and Component Models [Ref 2: Pg. 467-473]

10.3 Component Based Software Engineering (CBSE) Processes [Ref 2: Pg. 473-480]

10.3.1 CBSE for Reuse

10.3.2 CBSE with Reuse

10.4 Component Composition [Ref 2: Pg. 480-486]

11. System Modelling Using Case Studies (7 hours)

Prepare Object Oriented Analysis and Design documents for the Case Studies given in the VLE.

Teaching /Learning Methods:

You can access all learning materials and this syllabus in the VLE: <http://vle.bit.lk/>, if you are a registered student of the BIT degree program.

Assessment Strategy:**Continuous Assessments/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). The maximum mark for a question is 10 and the minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering both assignments, and students will have to obtain at least 40% for each assignment. Students are advised to complete online assignments before the given deadline. It is compulsory to pass all online assignments to qualify to obtain the Level II, Higher Diploma in IT (HDIT), certificate.

In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.

Final Exam:

Final examination of the course will be held at the end of the semester. The course is evaluated using a two hour question paper which consists of 25 MCQ (1 hour) and 2 Structured Questions (1 hour).

References/ Reading Materials:

Ref 1: Alan Dennis, Barbara Haley, David Tegarden, Systems analysis design, An Object Oriented Approach with UML : an object oriented approach, 5th edition, John Wiley & Sons, 2015, ISBN 978-1-118-80467-4

Ref 2: Ian Sommerville, Software Engineering, 10th edition, Pearson , Pearson Education, 2016, ISBN: 978-0133943030

Ref 3: <https://www.geeksforgeeks.org/computer-aided-software-engineering-case/>

Ref 4.: <https://www.uml-diagrams.org/>