



**Level II Curriculum 2021**  
**[Semester 3]**



**University of Colombo School of Computing**  
**December 2020**

## Table of Contents

EN3106 - Communication Skills II .....	3
IT 3106 - Object Oriented Analysis and Design .....	8
IT3206 - Data Structures and Algorithms .....	13
IT3306 - Data Management Systems .....	18
IT3406 - Web Application Development II .....	23

## EN3106 - Communication Skills II

<b>Semester</b>	<b>III</b>		
<b>Course Code:</b>	<b>EN3106</b>		
<b>Course Name:</b>	<b>Communication Skills II</b>		
<b>Credit Value:</b>	<b>2 (1L, 1P)</b>		
<b>Core/Optional</b>	<b>Core</b>		
<b>Hourly Breakdown</b>	<b>Theory</b>	<b>Practical</b>	<b>Independent Learning</b>
	15 hrs.	30Hrs	55 hrs.
<b>Course Aim/Intended Learning Outcomes (ILOs):</b>  At the completion of this course student will be able to: <ul style="list-style-type: none"> <li>Express concepts/ideas/proposals effectively to different audiences/stakeholders in a given context;</li> <li>Employ various tools and technologies in aid of effective professional and technical communication;</li> <li>Analyze the technical documents;</li> <li>Create well-organized technical reports that are structured according to acceptable standards;</li> <li>Describe how values and attitudes influence work management and communication;</li> <li>Apply practices required to be a significant team player in a collaborative working environment;</li> <li>Illustrate essential soft skills and technical skills for effective communicating within a team environment;</li> <li>Describe and critique different ways of conflict management to build stronger teams;</li> <li>Recognize the diversity in communication by giving due consideration to gender, culture, accessibility, digital divide, and economic issues in the society; and</li> <li>Demonstrate responsible behavior in the communication knowing the legal implications/consequences of irresponsibility and misconduct.</li> </ul>			
<b>Required Tools</b> <ul style="list-style-type: none"> <li>LaTeX, Word processing application (Microsoft Word, LibreOffice Writer, Apple Pages, etc.)</li> <li>Spreadsheet application (Microsoft Excel, LibreOffice Calc, Apple Numbers, etc.)</li> <li>Presentation applications (Microsoft PowerPoint, LibreOffice Impress, Apple Keynote, etc.)</li> <li>Google Docs, Git, Online meeting tools</li> <li>Google search</li> </ul>			

### Outline of the Syllabus

Topics	Theory (Hrs)	Practical (Hrs)
1. Introduction	1	-
2. Importance of Personal Development	1	1
3. Reading & Comprehension	2	3
4. Professional Responsibility, ethics and Plagiarism	1	2
5. Technical Writing and effective communication	5	12
6. Visual Communication	3	7
7. Team Work and Collaboration	2	5
<b>Total</b>	<b>15</b>	<b>30</b>

### Course Content:

#### 1. Introduction (1 hrs.)

- 1.1. Introduction to Communication Skills II [Ref: Teaching material]
- 1.2. Overview of the IT Industry [Ref: Teaching material]
  - 1.2.1. Career paths in IT
  - 1.2.2. Employability skills
- 1.3. Importance of Communication Skills in IT Industry [Ref: Teaching material]

#### 2. Importance of Personal Development (1 hrs.) [Ref: Teaching material]

- 2.1. Importance of interpersonal skills
- 2.2. Attitudes, behavior, and code of conduct
- 2.3. Personality and values

#### 3. Reading and comprehension (2 hrs.)

- 3.1. Introduction to different sources of reading [Ref 1: Pg. (4 - 11)]
  - 3.1.1. Technical Reports
  - 3.1.2. Business Communication
  - 3.1.3. Scientific Writings
- 3.2. Reading methods and techniques [Ref: Teaching material]
  - 3.2.1. Reading methods
  - 3.2.2. Reading techniques - skimming, scanning, intensive, extensive
  - 3.2.3. Mind maps

#### 4. Professional Responsibility ethics and Plagiarism (2 hrs)

- 4.1. Permissions and Plagiarism [Ref 1: Pg. (60 - 68)]
- 4.2. Using resources and their licenses [Ref 1: Pg. (174)] [Ref 5: Pg. (166-171)]
  - 4.2.1. Creative Commons [Ref 5: Pg. (98-113)]
  - 4.2.2. GNU Public License [Ref 5: Pg. (34-49)]
  - 4.2.3. Public Domain
- 4.3. Efficient use of Search Engines [Ref: Teaching material]

## **5. Technical Writing and effective communication (5 hrs.)**

### **5.1. Introduction to Technical Writing**

5.1.1. Introduction to Technical Writing [Ref 1: Pg. (1 - 3)]

5.1.2. Taxonomy of Technical Writing [Ref 1: Pg. (3 - 11)]

### **5.2. Technical Writing Basics**

5.2.1. Structuring your writing [Ref 1: Pg. (13 - 15)]

5.2.2. Know your audience [Ref 1: Pg. (16)]

5.2.3. Choosing the Right Words [Ref 1: Pg. (17 - 25)]

5.2.4. Avoiding Traps in Writing [Ref 1: Pg. (25 - 32)]

5.2.5. The 5Cs of Technical Writing [Ref 1: Pg. (36 - 41)]

5.2.5.1. Correctness

5.2.5.2. Clarity

5.2.5.3. Completeness

5.2.5.4. Consistency

5.2.5.5. Changeability

5.2.6. Referencing [Ref 1: Pg. (41 - 43)]

5.2.6.1. Referencing styles; IEEE and Harvard [Ref 7]

### **5.3. The Writing Process [Ref 1: Pg. (47 - 57)]**

5.3.1. Brainstorming

5.3.2. Drafting

5.3.3. Revising

5.3.4. Editing

5.3.5. Publishing

### **5.4. Writing Tools**

5.4.1. Introduction to Writing Tools [Ref 1: Pg. (59 - 60)]

5.4.2. LaTeX - A tool for technical writing [Ref 3: Pg. (1-41) (73-87), BIT Latex template]

### **5.5. Technical Reporting**

5.5.1. Systems Requirements Specification [BIT SRS guideline]

5.5.2. Technical Procedures [Ref 1: Pg. (135 - 143)]

5.5.3. User Manuals [Ref 1: Pg. (158 - 160)] [BIT Technical Manual guideline]

### **5.6. Preparing a Curriculum Vitae [Ref 1: Pg. (101 - 114)]**

### **5.7. BIT Dissertation [BIT dissertation guideline]**

## **6. Visual communication (3 hrs.)**

6.1. Using graphical elements in technical writing [Ref 1: Pg. (163 - 172)]

### **6.2. Choosing effective visuals**

6.2.1. Simple Text [Ref 2: Pg. (38 - 40)]

6.2.2. Tables [Ref 2: Pg. (40 - 43)] [Ref 1: Pg. (175-177)]

6.2.3. Graphs [Ref 2: Pg. (43 - 69)]

6.2.3.1. Points

6.2.3.2. Lines

<ul style="list-style-type: none"><li>6.2.3.3. Bars</li><li>6.2.3.4. Area</li><li>6.2.3.5. Graphs to be avoided</li><li>6.2.4. Presenting clean graphs - decluttering [Ref 2: Pg. (90-98)]</li><li>6.3. Figures [Ref 1: Pg. (172 - 175)]</li></ul> <p><b>7. Teamwork and Collaboration (2 hrs)</b></p> <ul style="list-style-type: none"><li>7.1. Introduction to Team Collaboration [Ref 4: Pg. (4-91)]</li><li>7.2. Team Management and Leadership [Ref 4: Pg. (93-152)]</li><li>7.3. Building Team Culture [Ref 4: Pg. (153-193)]</li><li>7.4. Avoiding Team Conflicts [Ref 4: Pg. (194-224)]</li><li>7.5. Collaboration tools<ul style="list-style-type: none"><li>7.5.1. Software Version Management and Collaboration [Ref 6: Chapter1 &amp; Chapter 2]</li><li>7.5.2. Online Collaboration Concepts and Tools [Ref: Teaching material]</li></ul></li></ul>
<p><b>Teaching /Learning Methods:</b></p> <p>Registered students of the BIT degree program can access learning materials and the syllabus in the VLE: <a href="http://vle.bit.lk">http://vle.bit.lk</a>. It is important to participate in learning activities given in the VLE to learn this subject.</p>
<p><b>Assessment Strategy:</b></p> <p><b>Continuous Assessments/Assignments:</b></p> <p>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 50% 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.</p> <p>In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.</p> <p><b>Final Exam:</b></p> <p>Final exam of the course will be held at the end of the semester. This course is evaluated using a one-hour written question paper which consists of 25 MCQs and an online assessment given in the VLE for two weeks soon after the written exam. In order to pass this enhancement course (a compulsory requirement to obtain the BIT degree), you will need to pass the online assessment AND the written exam paper in one sitting.</p>

**References/ Reading Materials:**

- Ref 1: Laplante, P. A., (2018) Technical Writing: A Practical Guide for Engineers, Scientists, and Nontechnical Professionals -Routledge; 2nd edition.
- Ref 2: Knafllic, C.N., (November 2, 2015) Storytelling with data: a data visualization guide for business professionals-Wiley; 1st edition.
- Ref 3: Oetiker, T., Partl, H., Hyna, I. and Schlegl, E., (1995). The not so short introduction to LATEX2 $\epsilon$ . Electronic document available at [http://www. tex. ac. uk/tex-archive/info/lshort](http://www.tex.ac.uk/tex-archive/info/lshort).
- Ref 4: Moga R., Olic A., The big book of team culture,
- Ref 5: Laurent, A.M.S., (2004) Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software. " O'Reilly Media, Inc."
- Ref 6: Chacon, S. and Straub, B., (2014). Pro git - Springer Nature.
- Ref 7: IEEE Referencing Guide, (2020)

**Supplementary Reading Materials:**

- Basson, A. H., von Backström, T. W., (2007), Guide for Writing Technical Reports
- Krause, S. D., Eastern Michigan University (2007), The Process of Research Writing
- Harvard Referencing Style, <https://www.mendeley.com/guides/harvard-citation-guide>
- <https://www.citethisforme.com/citation-generator/harvard>

## IT 3106 - Object Oriented Analysis and Design

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
<b>Course Aim:</b> To provide necessary skills and competencies to analyze and design a system using object-oriented approach		
<b>Intended Learning Outcomes:</b> After following this course, students should be able to <ul style="list-style-type: none"><li>- describe Object Oriented Analysis and Design concepts and apply them to solve problems</li><li>- define UML (Unified Modeling Language) and its various types of diagrams</li><li>- prepare Object Oriented Analysis and Design documents for a given problem using Unified Modeling Language</li><li>- explain the transition from analysis to design</li><li>- describe the key activities in the component-based software engineering (CBSE) process</li></ul>		
<b>Required Tools</b>  ArgoUML OR StarUML OR Visual Paradigm-Community Edition OR any tool that supports UML 1.4 and higher		
<b>Course Content: (Main Topics, Sub topics)</b>		
<b>Topic</b>	<b>Theory (Hrs)</b>	
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*	
<b>Total</b>	<b>45</b>	
<i>* Students are expected to do practical's and tutorials to strengthen their knowledge of these sections.</i>		



**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 Somerville, 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/>

## IT3206 - Data Structures and Algorithms

Semester	III	
Course Code:	IT3206	
Course Name:	Data Structures and Algorithms	
Credit Value:	3	
Core/Optional	Core	
Hourly Breakdown	Theory	Independent Learning
	45 hrs.	105 hrs.
<b>Course Aim:</b> This course module provides fundamental knowledge in the application of different data structures and algorithmic processes. Further, this provides knowledge in the sorting and searching algorithms. Finally provides skills to use data structures and algorithms for problem solving activities.		
<b>Intended Learning Outcomes:</b> After successfully completing this course, students should be able to <ul style="list-style-type: none"><li>• Explain common data structures and their applications</li><li>• Explain common searching and sorting algorithms and their applications</li><li>• Apply data structures and algorithms inappropriate real-world problem-solving activities.</li></ul>		
<b>Course Content: (Main Topics, Sub topics)</b>		
Topic	Theory (Hrs)	
1. Introduction to Data Structures	03	
2. Arrays and Linked Lists	05*	
3. Stacks and Queues	05*	
4. Recursion	04*	
5. Trees	08*	
6. Graphs	08*	
7. Analysis of Algorithms	05*	
8. Sorting and Searching Algorithms.	07*	
Total	45	
* Students are expected to do practical and tutorials to strengthen their knowledge of these sections		

### **1: Introduction to Data Structures (03 hours)**

#### **1.1. Introduction to data structures [Ref 3:pg.9, Ref 1:pg.230 ]**

**1.1.1. Real-world data storage [Ref 3:pg.10-11]**

**1.1.2. Programmer's Tools [Ref 3:pg.11]**

**1.1.3. Real-world Modeling [Ref 3:pg.11]**

#### **1.2. Use of Data Structures [Ref 3:pg.10-11]**

#### **1.3. Overview of Data Structures [Ref 3:pg.11-12]**

#### **1.4. Classification of Data Structures [Ref 4:pg.18]**

### **2: Arrays and Linked Lists (05 hours )**

#### **2.1. Introductuion to Arrays [Ref 1:pg.37-45, Ref 4:pg.75-77]**

**2.1.1. One dimensional arrays[Ref 1:pg.37-42, Ref 3:pg.39-42]**

**2.1.2. Multi dimensional arrays [Ref 1:pg.45]**

**2.1.3. Basic operations on arrays [Ref 3:pg.43-46]**

#### **2.2. Comparison of Arrays and Linked Lists [Ref 4:pg.76-78, pg.174]**

#### **2.3. Singly Linked Lists [Ref 1:pg.620-630, Ref 4:pg.78-88,]**

#### **2.4. Doubly Linked Lists [Ref 1:pg.630-633, Ref 4:pg.88-96, Ref 2:pg.236-239]**

#### **2.5. Circular Linked Lists [Ref 4:pg.96-107]**

#### **2.6. Skip Lists [Ref 4:pg.118-121]**

### **3: Stacks and Queues (05 hours)**

#### **3.1. Stacks**

**3.1.1. Intoduction to Stacks [Ref 4:pg.163-165,Ref 2:pg.232]**

**3.1.1.1. Array based Stack implementation [Ref 4:pg.165-167, Ref 3:pg. 116-123, Ref 1:pg.596-599, Ref 2:pg.233-234]**

**3.1.1.2. Linked List based Stack implementation [Ref 4:pg.171-175, Ref1:pg.606-609]**

**3.1.2. Applications of Stacks [Ref 4:pg.165, Ref 4:pg.174-185]**

#### **3.2. Queues**

**3.2.1. Introduction to queues [Ref 4:pg.205-206, Ref 2:pg.234]**

**3.2.1.1. Array based Queue implementation [Ref 1:pg.260-261, pg.600-604, Ref 3:pg.132-142, Ref 2:pg.235]**

**3.2.1.2. Linked List based Queue implementation [Ref 4: pg. 212-213, Ref1:pg.609-612]**

**3.2.2. Applications of queues [Ref 4:pg.207]**

**3.2.3. Circular queue [Ref 4:pg.207-208, Ref 3:pg.36-137]**

**3.2.4. Priority queue [Ref 4:pg.369-371, Ref 1:pg.274-276, Ref 3:pg.143-149]**

#### **4: Recursion ( 4 hours)**

- 4.1.** Introduction to Recursion [Ref 1:pg.294, Ref 4:pg. 62-65]
- 4.2.** Recursion versus iteration [Ref 4:pg.65-66]
- 4.3.** How recursion works [Ref 4:pg.66, Ref 1:pg.302-303]
- 4.4.** Implementation in recursion [Ref 4:pg.63-68, Ref 1:pg.304-314, Ref 5:pg.184-197]
  - 4.4.1.** Tail recursion [Ref 5:pg.184-185]
  - 4.4.2.** Nontail recursion [Ref 5:pg.185-191]
  - 4.4.3.** Indirect recursion [Ref 5:pg.191-193]
  - 4.4.4.** Nested Recursion [Ref 5:pg.193]
  - 4.4.5.** Excessive recursion [Ref 5:pg.194-197]

#### **5: Trees (08 hours)**

- 5.1.** Introduction to trees [Ref 1: pg.651-665, Ref 4: pg.224-231, Ref 3: pg.365-371]
  - 5.1.1.** Introduction to general trees, properties of general trees [Ref 1: pg.651-658, Ref 4: pg.224-227]
  - 5.1.2.** Introduction to binary trees, properties of binary trees [Ref 1: pg.658-665, Ref 4: pg.227-231, Ref 3: pg.365-371]
- 5.2.** Tree traversal [Ref 1: pg.667-679, Ref 4: pg.231-239, Ref 3: pg.381-388]
  - 5.2.1.** Depth First Traversal [Ref 1: pg.671-678, Ref 4: pg.231-238, Ref 3: pg.381-388]
    - 5.2.1.1.** Preorder Traversal
    - 5.2.1.2.** Inorder Traversal
    - 5.2.1.3.** Postorder Traversal
  - 5.2.2.** Breadth First Traversal (Level Order Traversal) [Ref 1: pg.678-679, Ref 4: pg.238-239]
- 5.3.** Binary search trees [Ref 1: pg.687-714, Ref 2: pg.286-289, Ref 4: pg.301-312, Ref 3: pg.371-376]
- 5.4.** Tree Balancing [Ref 1: pg.706-714, Ref 4: pg.329-342, Ref 5: pg.255-266]
  - 5.4.1.** Introduction to tree balancing [Ref 5: pg.255-258]
  - 5.4.2.** Global tree balancing (The DSW algorithm) [Ref 5: pg.258-261]
  - 5.4.3.** Local tree balancing (AVL Tree) [Ref 1: pg.706-714, Ref 4: pg.329-342, Ref 5: pg.258-261]
- 5.5.** Heaps [Ref 1: pg.808-818, Ref 4: pg.372-386, Ref 3: pg.580-601]

#### **6: Graphs (08 hours)**

- 6.1.** Introduction to graphs [Ref 1: pg.528-529, Ref 4: pg.426-432, Ref 3: pg.615-619]
  - 6.1.1.** Directed graphs, undirected graphs, weighted graphs [Ref 1: pg.528-529, Ref 4: pg.426-431, Ref 3: pg.615-619]
  - 6.1.2.** Applications of graphs [Ref 4: pg.432]
- 6.2.** Different types of Graph representations [Ref 1: pg.530-539, Ref 2: pg.589-592, Ref 4: pg.432-512, Ref 3: pg.619-622]
  - 6.2.1.** Array based implementation (Adjacency matrix, path matrix ) [Ref 4: pg.432-434, Ref 3: pg.620-621]
  - 6.2.2.** Linked-list based implementation (Adjacency list) [Ref 1: pg.530-539, Ref 4: pg.434-437, Ref 3: pg.621-622]

- 6.3. Graph traversal [Ref 2: pg.594-611, Ref 4: pg.438-449, Ref 3: pg.623-642]
  - 6.3.1. Depth first search (DFS) [Ref 2: pg.594-602, Ref 4: pg.438-445, Ref 3: pg.625-634]
  - 6.3.2. Breadth first search (BFS) [Ref 2: pg.603-611, Ref 4: pg.445-449, Ref 3: pg.636-642]
- 6.4. Shortest path algorithms graph [Ref 1: pg.539-554, Ref 2: pg.651-654, pg.658-662]
  - 6.4.1. Shortest path in unweighted graph [Ref 1: pg.539-545]
  - 6.4.2. Shortest path in weighted graph (Dijkstra's algorithm) [Ref 1: pg.545-552, Ref 2: pg.658-662]
  - 6.4.3. Shortest path weighted graph with negative edges (Bellman-Ford Algorithm) [Ref 1: pg.552-554, Ref 2: pg.651-654]

## **7: Analysis of Algorithms (05 hours)**

- 7.1. Introduction to analysis of algorithms [Ref 1: pg.188-192, Ref 4: pg.19-23, Ref 2: pg.23-29]
- 7.2. Types of analysis [Ref 4: pg.23-24, Ref 2: pg.43-44]
- 7.3. Big-O notation [Ref 1: pg.201-205, Ref 4: pg.24-26, Ref 2: pg.47-48]

## **8: Sorting and Searching Algorithms (07 hours)**

- 8.1. Introduction to iterative and, Divide and Conquer Methodology [Ref 2: pg.29-34, pg.65-67]
- 8.2. Sorting
  - 8.2.1. Iterative Method [Ref 2: pg.16-22, Ref 3: pg.79-103]
    - 8.2.1.1. Bubble sort [Ref 3: pg.79-89]
    - 8.2.1.2. Selection sort [Ref 3: pg.89-95]
    - 8.2.1.3. Insertion sort [Ref 2: pg.16-22, Ref 3: pg.95-103]
  - 8.2.2. Divide and Conquer Method [Ref 1: pg.361-369, pg.823-826, Ref 3: pg.279-294, pg.333-359, : pg.601-610, Ref 2: pg.151-169, pg.170-190]
    - 8.2.2.1. Merge sort [Ref 1: pg.361-364, Ref 3: pg.279-294]
    - 8.2.2.2. Quick sort [Ref 1: pg.364-369, Ref 3: pg.333-357, Ref 2: pg.170-190]
    - 8.2.2.3. Radix Sort [Ref 3: pg.357-359]
    - 8.2.2.4. Heap Sort [Ref 1: pg.823-826, Ref 3: pg.601-610, Ref 2: pg.151-169]
- 8.3. Searching algorithms
  - 8.3.1. Linear search [Ref 1: pg.207-208]
  - 8.3.2. Binary search [Ref 1: pg.208-210]
  - 8.3.3. Interpolation search [Ref 1: pg.211-212]

### **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:**

**Main references**

**Ref 1:** Data Structures and Problem solving using Java by Mark Allen Weiss, 4<sup>th</sup> Edition, ISBN 9780321541406

**Ref 2:** Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, The MIT Press, 3<sup>rd</sup> edition, ISBN 978-0262033848

**Ref 3:** Data Structures and Algorithms in Java by Robert Lafore, GC Join for Techmedia, 2<sup>nd</sup> edition, ISBN 978-0672324536

**Supplementary references**

**Ref 4:** Data Structures and Algorithms Made Easy, by Narasimha Karumanchi, 5<sup>th</sup> Edition, [Online source : <https://bit.ly/data-structures-and-algorithms-narasimha-karumanchi-pdf>]

**Ref 5:** Data Structures and Algorithms in Java by Adam Drozdek, Cengage Learning Asia, 3rd edition. ISBN 978-9814239233

**\*\*Note :** Ref 5 should be updated with 4<sup>th</sup> edition and page numbers should be updated accordingly.

## IT3306 - Data Management Systems

Semester	III	
Course Code:	IT3306	
Course Name:	Data Management Systems	
Credit Value:	3	
Core/Optional	Core	
Hourly Breakdown	Theory	Independent Learning
	45 hrs.	105 hrs.
<b>Course Aim:</b> <ul style="list-style-type: none"><li>To gain knowledge on different data management techniques with an insight on distributed database systems and other advanced database concepts.</li></ul>		
<b>Intended Learning Outcomes:</b> <p>After following this course, students should be able to,</p> <ul style="list-style-type: none"><li>Describe how and why different data management techniques evolved.</li><li>Apply constraints and to maintain consistency through triggers</li><li>Describe data storage &amp; improve query performance by using indexes.</li><li>Describe consistency and transaction processing techniques in different data management systems.</li><li>Describe distributed database concepts and create designs for distributed databases with an insight on how data fragmentation, replication and allocation affect database performance.</li></ul>		
<b>Course Content: (Main Topics, Sub topics)</b>		
Topic	Theory (Hrs)	
1. Data Management Evolution	08	
2. Database Constraints and Triggers	08 *	
3. Database Indexing and Tuning	10	
4. Distributed Database Systems	07	
5. Consistency and Transaction Processing Concepts	12*	
Total	45	
<i>*Students are expected to do practical and tutorials to strengthen their knowledge on these sections.</i>		

## **1. Data Management Evolution (8 hours)**

### **1.1. Major concepts of object-oriented, XML, and NoSQL databases**

#### **1.1.1. Object Databases**

- 1.1.1.1. Overview of Object Database Concepts
  - 1.1.1.1.1 Introduction to object-oriented concepts and features [Ref1: pg.363-366],
  - 1.1.1.1.2 Object Identity and Literals [Ref1: pg.367-370],
  - 1.1.1.1.3 Encapsulation of Operations [Ref1: pg. 370-372],
  - 1.1.1.1.4 Persistence of Objects [Ref1: pg. 373],
  - 1.1.1.1.5 Type Hierarchies and Inheritance [Ref1: pg. 374-376],
  - 1.1.1.1.6 Polymorphism and Multiple Inheritance [Ref1: pg. 377-378]

#### **1.1.2. XML Databases**

- 1.1.2.1. Reason for the origination of XML [Ref1: pg. 425, 426]
- 1.1.2.2. Structured, Semi structured, and Unstructured Data [Ref1: pg. 426-430]  
Structured data, Storage in relational database, Semi structured data, Directed graph model, Unstructured data
- 1.1.2.3. XML Hierarchical (Tree) Data Model [Ref1: pg. 430-433]  
Basic objects, Element, Attribute, Document types, Data-centric and Document-centric, Hybrid

#### **1.1.3. NoSQL Databases**

- 1.1.3.1. Origins of NoSQL  
Impedance Mismatch [Ref2: pg. 5-6], Problem of clusters[Ref2: pg. 8-9], Common characteristics of NoSQL databases [Ref2: pg. 9-10], Important rise of NoSQL with Polyglot Persistence [Ref2: pg. 11]
- 1.1.3.2. Data models in NoSQL
  - 1.1.3.2.1. Introduction to Aggregate data models [Ref2: pg. 13-19], Reason for using Aggregate data models [Ref2: pg. 19]
  - 1.1.3.2.2. Key-Value Model and suitable Use Cases [Ref2: pg. 20-21, Ref2: pg. 81-83, Ref2: pg.87-88]
  - 1.1.3.2.3. Document Data Model and suitable Use Cases [Ref2: pg.20-21, Ref2: pg. 89-91, Ref2: pg. 97-98]
  - 1.1.3.2.4. Column-Family Stores and suitable Use Cases [Ref2: pg. 21-23, Ref2: pg.99-102, Ref2: pg.107-109]
  - 1.1.3.2.5. Data model for complex relationship structures (Graph database model) [Ref2: pg. 25-28, Ref2: pg.111-114, Ref2: pg. 120-121]

### **1.2. Contrast and compare relational databases concepts and non-relational databases**

#### **1.2.1. Object databases and Relational databases** [Ref1: pg. 405-406]

#### **1.2.2. XML and Relational databases** [Ref1: pg. 425-426, Ref1: pg.430 - 436]

#### **1.2.3. NoSQL and Relational databases**

- 1.2.3.1. Data modeling difference [Ref2: pg. 13-19], Modeling for Data Access [Ref2: pg. 31-36]
- 1.2.3.2. Aggregate oriented vs aggregate ignorant [Ref2: pg.19-20]
- 1.2.3.3. Schemalessness in NoSQL [Ref2: pg. 10, Ref2: pg.28-30]
- 1.2.3.4. Overview of Materialized views [Ref2: pg. 30-31]

## **2. Database Constraints and Triggers [08 hours]**

### **2.1. Relational Model Constraints**

- 2.1.1. Categories of Constraints [Ref1: pg. 157-158]
- 2.1.2. Domain Constraints [Ref1: pg. 158]
  - 2.1.2.1. Key Constraints and Constraints on NULL Values [Ref1: pg. 158-160]
  - 2.1.2.2. Entity Integrity and Referential Integrity [Ref1: pg. 163-165]
  - 2.1.2.3. Other Types of Constraints [Ref1: pg. 165]
  - 2.1.2.4. Insert, Delete and Update Operations Dealing with Constraint Violations [Ref1: pg. 165-169]

### **2.2. Specifying Constraints in SQL**

- 2.2.1. Specifying Key and Referential Integrity Constraints [Ref1: pg. 184-187]
- 2.2.2. Specifying Constraints on Tuples Using CHECK [Ref1: pg. 187]
- 2.2.3. Specifying Names to Constraints [Ref1: pg. 185-187]

### **2.3. Constraints in Databases as Assertions [Ref1: pg. 225 -226]**

### **2.4. Specifying Actions in Databases as Triggers**

- 2.4.1. Introduction to Triggers and Create Trigger Statement [Ref1: pg. 225-227, pg. 966-967]
- 2.4.2. Active Databases and Triggers [Ref1: pg. 963-965]

## **3. Database Indexing and Tuning [10 hours]**

### **3.1. Disk Storage and Basic File Structures**

- 3.1.1. Computer memory hierarchy [Ref1: pg. 542-545],
- 3.1.2. Storage organization of databases [Ref1: pg. 545-546],
- 3.1.3. Secondary storage mediums [Ref1: pg. 547-551]
- 3.1.4. Solid State Device Storage [Ref1: pg. 553-554]
- 3.1.5. Placing file records on disk (types of records), [Ref1: pg. 560-564]
- 3.1.6. File Operations [Ref1: pg. 564-567]
- 3.1.7. Files of unordered records (Heap Files) and ordered records (Sorted Files) [Ref1: pg. 567-572]
- 3.1.8. Hashing techniques for storing database records: Internal hashing, external hashing [Ref1: pg. 572-575]

### **3.2. Introduction to indexing**

Introduce index files, indexing fields, index entry (record pointers and block pointers) [Ref1: pg. 601-602]

### **3.3. Types of Indexes**

- 3.3.1. Single Level Indexes: Primary, Clustering and Secondary indexes [Ref1: pg. 602 – 613]
- 3.3.2. Multilevel indexes: Overview of multilevel indexes [Ref1: pg. 613-616]

### **3.4. Indexes on Multiple Keys [Ref1: pg. 631-633]**

### **3.5. Other types of Indexes**

Hash indexes, bitmap indexes, function based indexes [Ref1: pg. 633-638]

### **3.6. Index Creation and Tuning [Ref1: pg. 639-641]**

### **3.7. Physical Database Design in Relational Databases [Ref1: pg. 643-646]**

**4. Distributed Database Systems [07 hours]**

**4.1. Distributed Database Concepts, Components and Advantages** [Ref1: pg. 841- 846]

**4.2. Types of Distributed Database Systems** [Ref1: pg. 865-866]

**4.3. Distributed Database Design Techniques** [Ref1: pg. 847-853]

4.3.1. Fragmentation [Ref1: pg. 847-849]

4.3.2. Replication and Allocation [Ref1: pg. 849-850]

4.3.3. Distribution Models: Single Server , Sharding, Master-Slave, Peer-to-Peer [Ref2: pg. 37-45]

**4.4. Query Processing and Optimization in Distributed Databases** [Ref1: pg. 859-862]

4.4.1 Distributed Query Processing

4.4.2 Data Transfer Costs of Distributed Query Processing

**4.5. NoSQL Characteristics related to Distributed Databases and Distributed Systems** [Ref1: pg. 885-886]

**5. Consistency and Transaction Processing Concepts [12 hours]**

**5.1. Introduction to Transaction Processing**

5.1.1. Single-user systems, multi-user systems and Transactions [Ref1: pg.746-747]

5.1.2. Problems in concurrent transaction processing, introduction to concurrency control, DBMS failures, introduction to data recovery [Ref1: pg.749-753]

5.1.3. Transaction states [Ref1: pg.753-754]

**5.2. Properties of Transactions**

5.2.1. ACID properties, levels of isolation [Ref1: pg. 757-758]

**5.3. Schedules**

5.3.1. Schedules of Transactions [Ref1: pg.759-760]

5.3.2. Schedules Based on Recoverability [Ref1: pg.761-762]

**5.4. Serializability**

5.4.1. Serial, Nonserial, and Conflict-Serializable Schedules [Ref1: pg.763-766]

5.4.2. Testing for Serializability of a Schedule [Ref1: pg.767-770]

5.4.3. Using Serializability for Concurrency Control [Ref1: pg.770 -771]

5.4.4. View Equivalence and View Serializability [Ref1: pg.771-772]

**5.5. Transaction Support in SQL** [Ref1: pg.773-776]

**5.6. Consistency in NoSQL**

5.6.1. Update Consistency [Ref2: pg.47-49]

5.6.2. Read Consistency [Ref2: pg.49-52]

5.6.3. Relaxing Consistency [Ref2: pg.52-53]

5.6.4. CAP theorem [Ref2: pg.53-56]

5.6.5. Relaxing Durability and Quorums [Ref2: pg.56-59]

5.6.6. Version Stamps [Ref2: pg.61-65]

**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. It is important to participate in learning activities given in the VLE to learn this course.

**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.** Fundamentals of Database Systems by R. Elmasri and S.B. Navathe, 7th edition, Addison-Wesley, 2016. ISBN : 978-0133970777

**Ref 2.** NoSQL Distilled: a brief guide to the emerging world of polyglot by Pramod J. Sadalage and Martin Fowler, 2013 Pearson Education, Inc.  
ISBN : 978-0321826626

**Note:** Please note that the page numbers given with respect to Ref 2 are different from the Ref 2 version which is downloadable from the internet.

## IT3406 - Web Application Development II

Semester	III		
Course Code:	IT3406		
Course Name:	Web Application Development II		
Credit Value:	4		
Core/Optional	Core		
Hourly Breakdown	Theory	Practical	Independent Learning
	45 hrs.	30 hrs.	125 hrs.
<b>Course Aim:</b> <ul style="list-style-type: none"><li>This module on web application development provides an insight to the server side web development technologies along with the advances features, methods and tools needed to add interactivity to produce rich internet applications.</li></ul>			
<b>Intended Learning Outcomes:</b> After successfully completing this course, students should be able to <ul style="list-style-type: none"><li>Describe and apply the fundamental and advanced concepts of PHP.</li><li>Describe the MVC architecture.</li><li>Employ Advanced features of client-side programming using JavaScript and Ajax to add interactivity to web pages.</li><li>Employ JavaScript libraries in web pages.</li><li>Describe and employ fundamental features of web security.</li></ul>			
<b>Course Content: (Main Topics, Sub topics)</b>			
Topic		Theory (Hrs)	Practical (Hrs.)
1. Server Side Web Development using PHP & MySQL		15	10
2. Fundamentals of Asynchronous JavaScript and XML (AJAX)		15	8
3. Advanced Client Side Development		9	8
4. Introduction to web application security		6	4
	Total	45	30
<b>1. Server Side Web Development using PHP &amp; MySQL (15 hours)</b>			
<b>Theoretical Aspects</b>			
1.1. Explain basic features of PHP [Ref 1: Pg. (303-304)] [Ref 10: Pg. (271-316)]			
1.2. Articulate and Explain MVC architecture [Ref 1: Pg. (683)] [Ref 3: Pg. (1-4)]			
1.3. Differentiate available PHP frameworks [Ref 3: Pg. (345-441)]/[Ref. 4: Pg. (83-105)]			
1.4. Consider server-side security threats [Ref 1: Pg. (375)] [Ref 10: Pg. (425-436)]			
1.4.1. Data spoofing [Ref 1: Pg. (376)]			
1.4.2. Invalid data [Ref 1: Pg. (379-380)]			
1.4.3.Unauthorized file access [Ref 1: Pg. (382)]			
1.5. Vulnerability solutions [Ref 1: Pg. (384-389)]			
1.5.1.Data sanitizing [Ref 1: Pg. (384)]			
1.5.2.Data validation [Ref 1: Pg. (389)]			

## **\*\*Guided Practicals**

- 1.1. Install PHP in a windows/ Linux environment [Ref 10: Pg. (40-54)]
- 1.2. Explain MVC architecture and differentiate available PHP frameworks [Ref 1: Pg. (683)] [Ref 3: Pg. (1-4)] [Ref 3: Pg. (345-441)]/[Ref. 4: Pg. (83-105)]
- 1.3. Explain basic features of PHP [Ref 1: Pg. (310-323, 325-343)] [Ref 10: Pg. (271-316)]
  - 1.3.1. PHP syntax and semantics
    - 1.3.1.1. Variables [Ref 1: Pg. (311)] [Ref 10 : Pg. (288-296)]
    - 1.3.1.2. Constants [Ref 10 : Pg. (287)]
    - 1.3.1.3. Conditional statements [Ref 1: Pg. (325-330)]
    - 1.3.1.4. Loops [Ref 1: Pg. (331-334)]
    - 1.3.1.5. Functions [Ref 1: Pg. (336)]
  - 1.3.2. Arrays and data processing with arrays [Ref 1: Pg. (206)] [Ref 10 : Pg. (296-305)]
  - 1.3.3. Handling HTML forms with GET and POST operations [Ref 1: Pg. (343)]
  - 1.3.4. Form validation fields ( including URLs and email address) and required fields [Ref 10: Pg. (574-585)]
  - 1.3.5. Filtering inputs ( validate and sanitize external inputs) [Ref 1: Pg. (384-389)] [Ref 10: Pg. (432)]
  - 1.3.6. Session control and cookies ( create and retrieve a cookie) PHP [Ref 1: Pg.(419-435)][Ref 10: Pg. (437-446)]
  - 1.3.7. File handling (Open, read, create, write operations with files, upload files ) PHP [Ref 10: Pg. (366-368)]
  - 1.3.8. Sending emails using PHP [Ref 11]
  - 1.3.9. Object Orientation with PHP [ Ref 1. Pg. (395-418)]
- 1.4. Use web services with PHP [Ref 10: Pg. (541-553)]
- 1.5. Develop a web application with PHP [Ref 9: Pg. (604-636)]
- 1.6. Setting up MySQL [Ref 1: Pg. (470-474)] [Ref 10: Pg. (56-74)]
- 1.7. Connect to MySQL database [Ref 1: Pg. (513)] [Ref 10: Pg. (515-527)] [Ref 2: Pg. (165)]
- 1.8. MYSQL database operations - Read/modify/delete/search operations [Ref 1: Pg. (497-513)] [Ref 2: Pg. (206)]
  - 1.8.1. Processing forms (Create, Read/Retrieve, Update, and Delete operations) [Ref 2: Pg. (235-250)]
- 1.9. Use of PHP unit testing tools for testing automation of front end web applications and manual testing of applications. [Ref 1: Pg. (616)]
- 1.10. Consider server-side security threats eg: data spoofing, invalid data, unauthorized file access [Ref 1: Pg. (375-382)][Ref 10: Pg. (425-436)]
- 1.11. Handle vulnerability solutions eg: data sanitizing and validation [Ref 1: Pg. (384-389)]

At the end of the section, students are guided to complete following **mini project**

- Create a basic PHP form that contains the following:
  - Consist of GET and POST operations.
  - Contains basic field validation, required field validation and validate email address, mobile number and URLs,.. etc
  - Once user submit button is pressed, how to keep the values of input field.
  - Process form by combining MySQL database to perform (Create, Read/Retrieve, Update, and Delete operations).
  - Test the front end of the PHP application using Selenium PHP unit testing automation tool.
  - Write test cases and carry out manual testing.



## 2. Fundamentals of Asynchronous JavaScript and XML (AJAX) (15 hours)

### Theoretical Aspects

- 2.1. Explain, and employ the traditional technologies used in web Application development [Ref 9: Pg. (1-10)]
- 2.2. Discuss and differentiate AJAX and Non-AJAX Applications [Ref 9: Pg. (405-420)]
  - 2.2.1. Discuss serialized data in a structured formats eg: XML/JSON both synchronous and asynchronous
- 2.3. Discuss industry standard tools and technologies for web development [Ref 3: Pg. (43-65)]
- 2.4. Discuss pros and cons of development frameworks for web development [Ref 1: Pg. (695-710)] [Ref 3: Pg. (83-109)]

### \*\*Guided Practicals

- 2.1. Explain and differentiate the traditional technologies used in web Application development [Ref 9: Pg. (1-10)]
- 2.2. Creating a Simple AJAX application [Ref 1: Pg. (619-643)] [Ref 9: Pg. (408-411)] [Ref 5] [Ref 8]
  - 2.2.1. Differentiate AJAX and Non-AJAX applications [Ref 9: Pg. (405-420)]
  - 2.2.2. Basic AJAX connection [Ref 12: Pg. (227-233)]
  - 2.2.3. Using jQuery AJAX library [Ref 1: Pg. (629-634)]
  - 2.2.4. Using XML in PHP [Ref 1: Pg. (636)]
  - 2.2.5. Using XML in JavaScript [Ref 1: Pg. (640)]
  - 2.2.6. Dynamic Hypertext Markup Language (DHTML)
- 2.3. Basic AJAX functionalities
  - 2.3.1. AJAX XMLHttpRequest [Ref 12: Pg. (230)] [Ref 12]
  - 2.3.2. AJAX Request [Ref 9: Pg. (491)] [Ref 13]
  - 2.3.3. AJAX Response [Ref 14]
- 2.4. Develop a webpage employing PHP, and AJAX [Ref 9: Pg. (408-417)]
- 2.5. Discuss industry standard tools and technologies for web development [Ref 3: Pg. (43-65)]
- 2.6. Use collaboration tools such as GitHub to work with a team on web [Ref 3: Pg. (1-21)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form,
  - show how a web page can communicate with a web server while a user type characters in an input field.
  - Fetch information from a database with AJAX.
  - Interactive communication with an XML file.
  - To handle interactive searches (live search where results are getting live while user types).
  - Demonstrate poll results without reloading the page.

### 3. Advanced Client Side Development (9 hours)

#### Theoretical Aspects

- 3.1. Use JavaScript to add new features to make more richer and compelling user interface on web pages [Ref 1: Pg. (195)]
- 3.2. Use JavaScript libraries to Manipulate Document Object Model [Ref 1: Pg. (223)]
- 3.3. Employ AJAX and JavaScript together in a website [Ref 1: Pg. (619)]
  - 3.3.1. Understand Single Page Application development
- 3.4. Consider client-side security threats [Ref 9: Pg. (317-318)]
- 3.5. Evaluate design and architecture of a web considering issues such as design pattern (including MVC), and tradeoff between redundancy, scalability, state management and search engine optimizations [Ref 1: Pg. (683-694)]

#### \*\*Guided Practicals

- 3.1. Understanding JavaScript basics [Ref 1: Pg. (195-222)] [Ref 2: Pg. (329-346)] [Ref 10: Pg. (187-190)]
  - 3.1.1. Understanding Document Object Model [Ref 1: Pg. (223)] [Ref 10 Pg: (208-217)]
    - 3.1.1.1. Working with nodes and elements [Ref 1: Pg. (223-233), (259-261)]
    - 3.1.1.2. Working with the Document Object Model [Ref 1: Pg. (238)]
- 3.2. Use JavaScript Libraries [Ref 10 Pg: (219-240)]
  - 3.2.1. Introduction to JavaScript libraries [Ref 10 Pg: (219-240)]
  - 3.2.2. Working with Events and Event Listeners [Ref 10 Pg: (241-254)]
  - 3.2.3. Understanding events [Ref 10 Pg: (241-254)]
  - 3.2.4. jQuery fundamentals [Ref 1 Pg: (243-247)] [Ref 10 Pg: (219-232)]
    - 3.2.4.1. Adding jQuery, reacting to events with JavaScript and jQuery [Ref 10: Pg. (219-259)]
  - 3.2.5. Working with Forms [Ref 10 Pg: (242-247)]
  - 3.2.6. Keyboard events [Ref 10 Pg: (247-254)]
- 3.3. Building a JavaScript program [Ref 10: Pg. (191-214)]
  - 3.3.1. Employ AJAX and JavaScript together in a website [Ref 1: Pg. (619)]
- 3.4. Handle client-side security threats [Ref 9: Pg. (317-318)]
  - 3.4.1. Validate data inputs on the client side [Ref 2: Pg. (371-387)] [Ref 9: Pg. (381-402)]
- 3.5. Express constraints involved in state management (cookies, query string, sessions) in the web [Ref 2: Pg. (287-304)] [Ref 9: Pg. (301-320)]
- 3.6. Employing development framework such as jQuery, Angular, ASP.NET MVC, WordPress etc. [Ref 1: Pg. (651-677, 695-713)] [Ref 10: Pg. (219-240)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form in section 2,
  - Add HTML form validations via Javascript.

#### 4. Introduction to web security ( 6 Hrs)

##### Theoretical Aspects

- 4.1. Produce secure web application by utilizing authentication, secure certificate, encryption, hashing, cookies and sessions. [Ref 2: Pg. (367-391)] [Ref 1: Pg. (419-428)]
  - 4.1.1. Setting, accessing and destroying a cookie [Ref 1: Pg. (419-428)]
  - 4.1.2. HTTP Authentication - Storing Usernames and passwords [Ref 4: Pg. (53-84)]
  - 4.1.3. Using sessions – Starting and ending sessions, session security and timeout [Ref 1: Pg. (430-435)] [Ref 9: Pg. (432-446)]
- 4.2. Able to discuss on common types of vulnerabilities and attacks in web [Ref 4: Pg. (11-19)]
  - 4.2.1. Injection [Ref 4: Pg. (13)]
  - 4.2.2. Cross-site scripting [Ref 4: Pg. (13)]
  - 4.2.3. Broken authentication and session management [Ref 4: Pg. (14)]
  - 4.2.4. Security misconfiguration [Ref 4: Pg. (16)]
  - 4.2.5. Insecure cryptographic storage [Ref 4: Pg. (16)]
  - 4.2.6. Failure to restrict URL access [Ref 4: Pg. (17)]
  - 4.2.7. Unvalidated redirects and forwards [Ref 4: Pg. (19)]
- 4.3. Differentiate client security and server security [Ref 9: Pg. (317-318)] [Ref 10: Pg. (425)]
  - 4.3.1. Securing server and client machines [Ref 10: Pg. (425)]
  - 4.3.2. Securing client application and apache web server [Ref 10: Pg. (425-426)]
  - 4.3.3. Configure PHP securely [Ref 10: Pg. (425-426)]
  - 4.3.4. Handling errors safely [Ref 10: Pg. (429-431)]
  - 4.3.5. Sanitizing variables [Ref 10: Pg. (432-434)]

##### \*\*Guided Practicals

- 4.1. Illustrate browser security models including same-origin policy and threat models in web security [Ref 4: Pg. (35-44), Ref 4: Pg. (149-155)]
- 4.2. Employ how authentication, secure certificates, secure communication - SSL can be used in web sessions [Ref 1: Pg. (437-447)] and common type of vulnerabilities [Ref 4: Pg. (11-19)]
  - 4.2.1. Setting, accessing and destroying a cookie [Ref 1: Pg. (419-428)]
  - 4.2.2. HTTP Authentication - Storing Usernames and passwords [Ref 4: Pg. (53-84)]
  - 4.2.3. Using sessions – Starting and ending sessions, session security and timeout [Ref 1: Pg. (430-435)] [Ref 9: Pg. (432-446)]
- 4.3. How sessions, cookies, encryption and hashing provide means of secure web application [Ref 2: Pg. (287-290, 290-296, 299-304)] [Ref 4: Pg. (367-391)]
  - 4.3.1. Using Cookies in PHP [Ref 2: Pg. (287-290)]
    - 4.3.1.1. Setting a Cookie [Ref 2: Pg. (289)]
    - 4.3.1.2. Accessing a Cookie [Ref 2: Pg. (290)]
    - 4.3.1.3. Destroying a Cookie [Ref 2: Pg. (290)]
  - 4.3.2. HTTP Authentication - Storing Usernames and passwords [Ref 2: Pg. (290-296)]
  - 4.3.3. Using Sessions [Ref 2: Pg. (299-304)]
    - 4.3.3.1. Starting a Session [Ref 2: Pg. (299)]
    - 4.3.3.2. Ending a Session [Ref 2: Pg. (299)]
    - 4.3.3.3. Setting a Timeout [Ref 2: Pg. (303)]
    - 4.3.3.4. Session Security [Ref 2: Pg. (304)]
  - 4.3.4. Hashing by using SHA1, MD5 [Ref 6,7]
- 4.4. Difference between http and https roles in web [Ref 4: Pg. (18-19)]

4.5. Handle client and server security [Ref 9: Pg. (317-318)] [Ref 10: Pg. (425)]

4.5.1. Configure PHP securely [Ref 10: Pg. (425-426)]

4.5.2. Handling errors safely [Ref 10: Pg. (429-431)]

4.5.3. Sanitizing variables [Ref 10: Pg. (432-434)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form in section 2,
  - Include sessions, cookies and suitable hashing techniques to make the system more secure.

**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:**

**Main references**

- **Ref 1.** PHP, MySQL, & JavaScript All-in-One For Dummies Richard Blum, 2017 (Online source : <https://www.pdfdrive.com/php-mysql-javascript-all-in-one-for-dummies-e90592496.html> )
- **Ref 2.** Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 5th Edition, O'Reilly Media, Inc. 2018 (online source : <https://www.pdfdrive.com/learning-php-mysql-javascript-with-jquery-css-html5-e188490793.html> ) .

**Supplementary references**

- **Ref 3.** PHP Development Tool Essentials, Chad Russell, 2016 (online source: <https://ikamy.ch/public/img/books/PHP+Development+Tool+Essentials.pdf>)
- **Ref 4.** Web Application Security, A Beginner's Guide McGraw-Hill Education; by Bryan Sullivan and Vincent Liu, 1st Edition (2011)
- **Ref 5.** [https://www.w3schools.com/php/php\\_ajax\\_intro.asp](https://www.w3schools.com/php/php_ajax_intro.asp)
- **Ref 6.** [https://www.w3schools.com/php/func\\_string\\_md5.asp](https://www.w3schools.com/php/func_string_md5.asp)

- Ref 7. [https://www.w3schools.com/PHP/func\\_string\\_sha1.asp](https://www.w3schools.com/PHP/func_string_sha1.asp)
- Ref 8. <https://www.w3schools.com/php/>
- Ref 9. Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition, O'Reilly Media, Inc. 2012.
- Ref 10. PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies , John Wiley & Sons, Inc. 2013
- Ref 11. [https://www.w3schools.com/php/php\\_form\\_url\\_email.asp](https://www.w3schools.com/php/php_form_url_email.asp)
- Ref 12 JavaScript and AJAX For Dummies, Andy Harris, 2009
- Ref 13. [https://www.w3schools.com/xml/ajax\\_xmlhttprequest\\_create.asp](https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp)
- Ref 14. [https://www.w3schools.com/xml/ajax\\_xmlhttprequest\\_send.asp](https://www.w3schools.com/xml/ajax_xmlhttprequest_send.asp)
- Ref 15. [https://www.w3schools.com/xml/ajax\\_xmlhttprequest\\_response.asp](https://www.w3schools.com/xml/ajax_xmlhttprequest_response.asp)