



# Diploma in Information Technology Program

## Information and Communication Technology Center

### Wayamba University of Sri Lanka

---

#### 1. Program Title

Diploma in Information Technology (DIT)  
Equivalent to the SLQF level 3.

#### 2. Program Overview

The Diploma in Information Technology (DIT) program offers students a comprehensive foundation in the field of information technology, preparing them for entry-level positions in various industries or further study in the field. This program covers essential topics in IT, including computer applications, networking, programming, database management, security, web development and IT project management. Through a combination of theoretical knowledge and hands-on practical experience, students will develop the skills and competencies necessary to succeed in today's rapidly evolving technology landscape.

#### 3. Program-level Objectives

This diploma in information technology program is designed to achieve the following program-level objectives.

##### i. Provide a Strong Foundation for Further Studies in Computing:

Offer comprehensive instruction in fundamental IT concepts, principles, and practices to prepare students for advanced studies in computing or related fields, equipping them with the necessary knowledge and skills for academic success in higher education.

##### ii. Enable Entry into the IT Workforce at the Entry Level:

Equip students with practical skills and industry-relevant knowledge to enter the IT workforce at the entry level, enabling them to pursue career opportunities in various sectors such as software development, IT support, system administration, and network management.

##### iii. Offer Opportunities to Learn Computer Fundamentals for Job Enhancement:

Provide working professionals with an opportunity to acquire essential computing knowledge and skills necessary to enhance their job performance and career prospects, empowering them to effectively utilize technology in their respective job roles and stay competitive in today's digital workplace.

##### iv. Foster Critical Thinking and Problem-Solving Abilities:

Cultivate students' critical thinking and problem-solving abilities through hands-on learning experiences and real-world IT scenarios, enabling them to analyze complex problems, devise effective solutions, and apply technical knowledge to practical challenges encountered in the IT industry.

- v. Promote Ethical and Professional Conduct in IT Practices:  
Emphasize the importance of ethical behavior, professional conduct, and responsible use of technology in IT practices, instilling values of integrity, accountability, and respect for privacy and security, ensuring that graduates uphold ethical standards and contribute positively to the IT profession and society.

#### 4. Entry Criteria

To be eligible for admission to the Diploma in Information Technology program, candidates must meet the following entry criteria:

- i. General Certificate of Education (Advanced level) or an equivalent qualification,  
OR
- ii. A foundation course equivalent to SLQF level 2 after a minimum of 12 years of schooling followed by passing the aptitude test given by the university,  
OR
- iii. Completion of NVQF level 4 or accredited work experience or accredited prior learning followed by a corresponding cognitive bridging programme of minimum 30 credits as determined by the university.  
OR
- iv. Passed the General Certificate of Education (Ordinary level) examination with Credit pass in English and Mathematics, followed by passing the aptitude test given by the university,

#### 5. Program Structure

The Diploma in Information Technology program is structured over a duration of one year and consists of two semesters. The program curriculum is designed to provide students with a balanced blend of theoretical knowledge and practical skills development. The modules of the program include:

<b>Year1 Semester1</b>			
Course Code	Course Name	Credits	Compulsory
DIT11011	Introduction to Computer Applications	1	Yes
DIT11022	Computer Systems and Architectures	2	Yes
DIT11033	System Analysis and Design	3	Yes
DIT11043	Introduction to Database Systems	3	Yes
DIT11052	Mathematics for Computing	2	Yes
DIT11064	Introduction to Web Technologies	4	Yes
	Sub total	15	

<b>Year 1 Semester2</b>			
Course Code	Course Name	Credits	Compulsory
DIT12074	Introduction to Programming	4	Yes
DIT12083	Object Oriented Concepts	3	Yes
DIT12093	Networking and Security Essentials	3	Yes
DIT12105	IT Project Development	5	Yes
	Sub total	15	
	<b>Total</b>	<b>30</b>	

6. Module outlines

<b>Course Code</b>	<b>DIT11011</b>			<b>Course Title</b>	<b>Introduction to Computer Applications</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>1</b>	<b>Credit</b>	<b>01</b>	<b>Theory (hr)</b>	15
						<b>Practical (hr)</b>	00
						<b>Independent Learning (hr)</b>	35
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		
<b>Aim of the Course:</b>							
<p>The aim of this course is to provide students with a foundational understanding of key computer applications and technologies essential for efficient digital productivity. Students will gain practical skills in using computers, operating systems, word processing, spreadsheets, presentation software, and web basics. The course aims to empower students to leverage these technologies effectively in academic, professional, and personal contexts.</p>							
<b>Intended Learning Outcomes:</b>							
<p>By the end of the course, students should be able to:</p> <p>LO1: Demonstrate proficiency in using computers and operating systems for basic tasks such as file management and system navigation.</p> <p>LO2: Utilize word processing software to create, format, and manage documents with advanced techniques including tables, graphics, and collaboration features.</p> <p>LO3: Apply spreadsheet fundamentals to organize data, perform calculations, and create visual representations for analysis and reporting.</p>							
<b>Content:</b>							
<p>Topic 1: Introduction to Computers and Operating Systems</p> <ul style="list-style-type: none"> <li>• Overview of Computers and their Components</li> <li>• Introduction to Operating Systems (Windows/Mac/Linux)</li> <li>• Basic File Management (Creating, Saving, Organizing Files)</li> </ul>							
<p>Topic 2: Word Processing and Document Management</p> <ul style="list-style-type: none"> <li>• Introduction to Word Processing (Microsoft Word, Google Docs)</li> <li>• Formatting Text and Documents</li> <li>• Document Layout and Design</li> <li>• Advanced Word Processing Techniques</li> <li>• Tables, Graphics, and Hyperlinks</li> <li>• Document Collaboration and Sharing</li> </ul>							
<p>Topic 3: Spreadsheet Fundamentals</p> <ul style="list-style-type: none"> <li>• Introduction to Spreadsheets (Microsoft Excel, Google Sheets)</li> <li>• Data Entry and Formatting</li> <li>• Basic Formulas and Functions</li> <li>• Advanced Spreadsheet Techniques</li> <li>• Charts and Graphs</li> <li>• Data Analysis with PivotTables</li> </ul>							
<p>Topic 4: Presentation Software and Graphics</p> <ul style="list-style-type: none"> <li>• Introduction to Presentation Software (Microsoft PowerPoint, Google Slides)</li> <li>• Designing Effective Slides</li> </ul>							

- Adding Multimedia Elements
- Advanced Presentation Techniques
- Slide Transitions and Animations
- Presenting and Sharing Slideshows

#### Topic 5: Web Basics and Overview

- Introduction to Web Browsers (Chrome, Firefox)
- Navigating the Web and Search Engines
- Understanding URLs and Hyperlinks
- Website Creation Basics (WordPress, Wix, or simple HTML/CSS)
- Web Hosting and Domain Names

#### **Assessment Criteria:**

Assessment will be based on the following criteria:

Practical Test 1 – 50%

Document Creation: Quality and complexity of documents created using word processing software, including formatting, layout, and collaboration features.

Practical Test 2 – 50%

Spreadsheet Proficiency: Ability to input, format, and analyze data using spreadsheet software, including the use of formulas, functions, and charts.

#### **Recommended Reading:**

- Microsoft Office 365 & Office 2019: Introductory by Misty E. Vermaat
- Excel 2021 By Jiayi Simonds, ISBN: 9798201437633
- PowerPoint For Dummies by Doug Lowe, ISBN: 9781119829140

<b>Course Code</b>	<b>DIT11012</b>			<b>Course Title</b>	<b>Computer Systems and Architectures</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>1</b>	<b>Credit</b>	<b>02</b>	<b>Theory (hr)</b>	20
						<b>Practical (hr)</b>	20
						<b>Independent Learning (hr)</b>	60
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

The aim of this introductory course in Computer Systems and Architecture is to provide foundational knowledge to beginners about how computers work at a hardware level. Students will gain a fundamental understanding of computer components, architecture, and system organization, preparing them for further studies in computer science and related fields.

### Intended Learning Outcomes:

LO1: Understand Computer Components and Architecture: Describe the basic components of a computer system and explain the Von Neumann architecture model.  
LO2: Explain CPU Functionality and Instruction Set Architecture: Identify the functions of the CPU, differentiate between CISC and RISC architectures, and understand basic instruction set concepts.  
LO3: Comprehend Memory Hierarchy and Storage Systems: Define different memory types, explain memory organization including registers and cache, and discuss storage technologies like HDD, SSD, and RAID.

### Content:

#### Topic 1: Introduction to Computer Systems

- Overview of Computer Architecture
- Basic Components: CPU, Memory, Input/Output Devices
- Von Neumann Architecture

#### Topic 2: CPU and Instruction Set Architecture

- Functions of the CPU (Arithmetic, Logic, Control)
- Instruction Set Architecture (ISA)
- Different CPU Architectures (CISC vs. RISC)

#### Topic 3: Memory Hierarchy

- Memory Types: RAM, Cache, Secondary Storage
- Memory Organization (Registers, Cache Levels)
- Cache Memory: Concepts and Performance

#### Topic 4: Input/Output and Peripheral Devices

- I/O Devices: Types and Functions
- I/O Controllers and Interfaces
- Interrupts and DMA (Direct Memory Access)

#### Topic 5: Computer Arithmetic

- Number Systems: Binary, Decimal, Hexadecimal
- Arithmetic Operations in Binary
- Floating-Point Representation and Arithmetic

#### Topic 6: Assembly Language and Machine Code

- Introduction to Assembly Language Programming
- Basic Instructions and Addressing Modes
- Assembly vs. High-Level Languages

Topic 7: Computer Organization

- Data Path and Control Unit
- Fetch-Decode-Execute Cycle
- Pipelining and Parallel Processing

Topic 8: System Bus and Interconnection

- System Bus Architecture (Address Bus, Data Bus, Control Bus)
- Bus Topologies: Parallel vs. Serial
- Interconnection Networks

Topic 9: Memory Management

- Memory Addressing Modes
- Virtual Memory Concepts
- Paging and Segmentation

Topic 10: Storage Systems

- Disk Storage Technology (HDD vs. SSD)
- RAID (Redundant Array of Independent Disks)
- File Systems: FAT, NTFS, EXT4, etc.

Topic 11: Introduction to Operating Systems

- OS Functions and Services
- OS Kernel and User Space
- Process Management and Scheduling

Topic 12: Review and Discussion

- Recap of Key Concepts
- Real-World Applications and Examples

**Assessment Criteria:**

LMS based assessment (100%):

Assess understanding of computer components, architecture, CPU functionality, and instruction set architecture through regular quizzes / assignments.

**Recommended Reading:**

- “Computer Systems: A Programmer's Perspective” by Randal E. Bryant and David R. O'Hallaron, ISBN: 9781292101774
- “Fundamentals of Information and Communication Technology” by Jayathilake C. ISBN 9786245720262
- “Structured Computer Organization” by Andrew S. Tanenbaum and Todd Austin, ISBN: 9780132916523

Course Code	DIT11033			Course Title	System Analysis and Design			
Level	3	Semester	1	Credit	03	Theory (hr)	30	
						Practical (hr)	30	
						Independent Learning (hr)	90	
Status	Core Mandatory			Pre-requisite Course/s	-			
<b>Aim of the Course:</b>								
The aim of this introductory course is to provide beginners with a foundational understanding of system analysis and design concepts and methodologies, preparing them for entry-level roles in the field.								
<b>Intended Learning Outcomes:</b>								
LO1: Define systems and systems analysis, explain the role of systems analysts, and describe the stages of the system development life cycle. LO2: Demonstrate the ability to gather stakeholder requirements using various elicitation techniques, analyze requirements, and document them effectively. LO3: Apply practical skills in modeling system requirements, designing system architecture, and implementing software design patterns for system development.								
<b>Content:</b>								
Topic 1: Introduction to System Analysis and Design <ul style="list-style-type: none"> <li>• Definition of Systems and Systems Analysis</li> <li>• Role of Systems Analysts</li> <li>• Overview of System Development Life Cycle (SDLC)</li> </ul>								
Topic 2: Requirements Gathering and Analysis <ul style="list-style-type: none"> <li>• Understanding Stakeholders and User Requirements</li> <li>• Techniques for Requirements Elicitation (Interviews, Questionnaires, Observations)</li> <li>• Requirements Analysis and Documentation</li> </ul>								
Topic 3: Modeling System Requirements <ul style="list-style-type: none"> <li>• Introduction to Modeling Techniques (Use Case Diagrams, Data Flow Diagrams)</li> <li>• Use Case Modeling: Actors, Use Cases, Relationships</li> <li>• Data Modeling: Entity-Relationship Diagrams (ERD)</li> </ul>								
Topic 4: System Design Principles <ul style="list-style-type: none"> <li>• Overview of System Design Process</li> <li>• Designing System Architecture (High-level and Detailed Design)</li> <li>• Interface Design and Prototyping</li> </ul>								
Topic 5: Designing Databases and Data Structures <ul style="list-style-type: none"> <li>• Database Design Concepts (Normalization, Schema Design)</li> <li>• Designing Data Structures for Efficient Information Processing</li> <li>• Introduction to Database Management Systems (DBMS)</li> </ul>								
Topic 6: Software Design Patterns <ul style="list-style-type: none"> <li>• Understanding Software Design Patterns (Creational, Structural, Behavioral)</li> <li>• Applying Design Patterns to System Design</li> <li>• Importance of Reusability and Modularity in Design</li> </ul>								

Topic 7: User Interface Design

- Principles of User Interface (UI) Design
- Designing User-friendly Interfaces
- Usability Testing and Evaluation

Topic 8: System Implementation Strategies

- Overview of System Implementation Phase
- Choosing Development Platforms and Technologies
- Coding Standards and Best Practices

Topic 9: Testing and Quality Assurance

- Software Testing Fundamentals (Types of Testing: Unit, Integration, System, Acceptance)
- Importance of Quality Assurance in System Development
- Test Planning and Execution

Topic 10: Deployment and Maintenance

- Deployment Strategies (Phased Rollout, Big Bang)
- System Maintenance and Upgrades
- Handling Change Requests and Version Control

Topic 11: Project Management for System Development

- Introduction to Project Management Methodologies (Agile, Waterfall, etc.)
- Managing Project Scope, Time, and Resources
- Risk Management in System Development

Topic 12: Review and Discussion

- Recap of Key Concepts
- Case Studies and Real-World Examples

**Assessment Criteria:**

LMS based assessment (40%):

Definitions of systems and systems analysis, role of systems analysts, overview of SDLC. Requirements gathering techniques, modeling system requirements, system design principles.

Final Examination (60%):

A comprehensive examination covering topics such as database design, software design patterns, and project management, assessing students' understanding and application of concepts learned throughout the course.

**Recommended Reading:**

- “Systems Analysis and Design” by Alan Dennis, Barbara Haley Wixom, and Roberta M. Roth, ISBN: 9781119585855
- “Database Systems: A Practical Approach to Design, Implementation, and Management” by Thomas Connolly and Carolyn Begg



<b>Course Code</b>	<b>DIT11043</b>			<b>Course Title</b>	<b>Introduction to Database Systems</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>1</b>	<b>Credit</b>	<b>03</b>	<b>Theory (hr)</b>	30
						<b>Practical (hr)</b>	60
						<b>Independent Learning (hr)</b>	60
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

To provide beginners with a foundational understanding of databases, covering fundamental concepts, SQL queries, relational database management systems (RDBMS), database design, and basic administration.

### Intended Learning Outcomes:

LO1: Define databases, explain the importance of databases in information management, and describe the components and benefits of database management systems (DBMS).  
 LO2: Demonstrate understanding by interpreting relational data model basics, SQL queries, and database design fundamentals, including normalization and entity-relationship (ER) diagrams.  
 LO3: Apply their knowledge by writing SQL queries, setting up RDBMS environments, querying multiple tables using joins, and implementing basic database administration tasks.

### Content:

#### Topic 1: Introduction to Databases

- What is a Database?
- Importance of Databases in Information Management
- Overview of Database Management Systems (DBMS) and their benefits

#### Topic 2: Relational Data Model Basics

- Understanding Tables, Rows, and Columns
- Introduction to Entities, Attributes, and Relationships
- Overview of Primary Keys and Foreign Keys

#### Topic 3: Introduction to SQL (Structured Query Language)

- Basics of SQL: SELECT statement for retrieving data
- Filtering data using WHERE clause
- Sorting data using ORDER BY clause

#### Topic 4: More SQL Queries

- Using SQL functions (e.g., COUNT, SUM, AVG)
- Performing data manipulation with INSERT, UPDATE, and DELETE statements

#### Topic 5: Database Design Fundamentals

- Basics of Database Design and Normalization
- Understanding Entity-Relationship (ER) diagrams
- Creating simple ER diagrams for given scenarios

#### Topic 6: Introduction to Relational Database Management Systems (RDBMS)

- Overview of popular RDBMS (e.g. MySQL)
- Installing and setting up a local database environment

Topic 7: Querying Multiple Tables (Joins)

- Understanding JOINS (INNER JOIN, LEFT JOIN, RIGHT JOIN)
- Writing SQL queries to retrieve data from multiple related tables

Topic 8: Data Integrity and Constraints

- Importance of Data Integrity in databases
- Overview of Primary Key, Foreign Key, and UNIQUE constraints

Topic 9: Basic Database Administration

- Managing database objects (tables, indexes, views)
- Backing up and restoring databases

Topic 10: Introduction to NoSQL Databases

- Overview of NoSQL databases (e.g., MongoDB, Redis)
- Understanding key-value stores and document stores

Topic 11: Database Security Basics

- Introduction to Database Security Principles
- User authentication and authorization

Topic 12: Review and Q&A

- Recap of key concepts covered in the course
- Open forum for questions and discussion

**Assessment Criteria:**

Quizzes (30%):

Introduction to databases, relational data model basics, SQL basics.

Practical Exam (50%):

SQL queries, database design fundamentals, RDBMS setup, querying multiple tables, basic database administration.

Final Examination (20%):

A comprehensive examination covering topics such as database security, NoSQL databases, and a review of key concepts learned throughout the course.

**Recommended Reading:**

- “Database Systems: The Complete Book” by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom, ISBN: 9788131708422
- “SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL” by John L. Viescas and Michael J. Hernandez
- “Learning SQL: Generate, Manipulate, and Retrieve Data” by Alan Beaulieu, ISBN: 9781492057581

<b>Course Code</b>	<b>DIT11052</b>			<b>Course Title</b>	<b>Mathematics for Computing</b>		
<b>Level</b>	3	<b>Semester</b>	1	<b>Credit</b>	02	<b>Theory (hr)</b>	30
						<b>Practical (hr)</b>	15
						<b>Independent Learning (hr)</b>	55
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to provide students with a solid foundation in essential mathematical concepts and techniques relevant to computing, enabling them to apply mathematical principles effectively in various computing contexts.

### Intended Learning Outcomes:

LO1: Demonstrate understanding of fundamental mathematical concepts including numbers, algebra, logic, geometry, sets, probability, statistics, graphs, combinatorics, linear algebra, and calculus.  
LO2: Apply mathematical techniques to solve problems in computing, analyze data, and make informed decisions based on statistical information and computational processes.

### Content:

#### Topic 1: Introduction to Numbers and Basic Operations

- Overview of Number Systems: Decimal, Binary, and Hexadecimal
- Arithmetic Operations: Addition, Subtraction, Multiplication, Division

#### Topic 2: Introduction to Algebra

- Understanding Variables and Expressions
- Solving Simple Equations (e.g.,  $2x + 3 = 7$ )
- Basic Concepts of Inequalities

#### Topic 3: Introduction to Logic

- Basic Logic Gates (AND, OR, NOT)
- Truth Tables and Logical Operations
- Boolean Algebra Basics

#### Topic 4: Introduction to Geometry

- Basic Geometric Shapes: Points, Lines, and Angles
- Properties of Triangles and Quadrilaterals
- Introduction to Cartesian Coordinates

#### Topic 5: Introduction to Sets and Venn Diagrams

- Understanding Sets and Set Operations (Union, Intersection, Complement)
- Visualizing Sets with Venn Diagrams

#### Topic 6: Introduction to Probability

- Basic Probability Concepts (Outcomes, Events, Probability of Events)
- Simple Probability Calculations

#### Topic 7: Introduction to Statistics

- Overview of Descriptive Statistics: Mean, Median, Mode

- Basic Data Analysis and Interpretation

Topic 8: Introduction to Graphs

- Understanding Graphs (Vertices and Edges)
- Basic Graph Algorithms (Traversal)

Topic 9: Introduction to Combinatorics

- Counting Principles: Permutations and Combinations
- Basic Combinatorial Problems and Applications

Topic 10: Introduction to Linear Algebra

- Basics of Vectors and Vector Operations
- Applications of Vectors in Computing

Topic 11: Introduction to Calculus Concepts

- Understanding Rates of Change and Slopes
- Basic Differentiation and Integration (Intuitive Concepts)

Topic 12: Review and Practical Applications

- Recap of Key Concepts Covered
- Practical Examples of Mathematics in Computing

**Assessment Criteria:**

LMS based assessment (50%):

Quizzes and assignments will assess students' understanding of key concepts and their ability to apply mathematical techniques. Quizzes will cover specific topics such as numbers and basic operations, algebra, logic, geometry, etc.

Final Examination (50%):

The final examination will comprehensively cover all topics taught throughout the course, including calculus concepts and practical applications.

**Recommended Reading:**

- "Mathematics for Computer Science" by Eric Lehman, F. Thomson Leighton, and Albert R. Meyer
- "Discrete Mathematics and Its Applications" by Kenneth H. Rosen, ISBN: 9781260091991
- "Linear Algebra and Its Applications" by David C. Lay, Steven R. Lay, and Judi J. McDonald, ISBN: 9780321982384

<b>Course Code</b>	<b>DIT11064</b>			<b>Course Title</b>	<b>Introduction to Web Technologies</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>1</b>	<b>Credit</b>	<b>04</b>	<b>Theory (hr)</b>	20
						<b>Practical (hr)</b>	60
						<b>Independent Learning (hr)</b>	120
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to provide students with a comprehensive understanding of web development fundamentals, including HTML, CSS, JavaScript, and essential web technologies, equipping them with the skills necessary to create and deploy interactive and responsive websites.

### Intended Learning Outcomes:

LO1: Demonstrate an understanding of the core concepts and technologies used in web development, including HTML structure, CSS styling, JavaScript interactivity, and responsive design principles.  
LO2: Apply their knowledge of HTML, CSS, and JavaScript to create functional and visually appealing web pages, incorporating interactive elements and responsive layouts.  
LO3: Develop problem-solving skills through hands-on projects and exercises, allowing them to troubleshoot issues, debug code, and effectively utilize web development tools for designing, testing, and deploying websites.

### Content:

#### Topic 1: Introduction to the World Wide Web

- Overview of the Internet and the Web
- Understanding Client-Server Architecture
- Evolution of Web Technologies

#### Topic 2: Basics of HTML (Hypertext Markup Language)

- Introduction to HTML
- Structure of HTML Documents: Elements, Tags, Attributes
- Creating a Simple Web Page with HTML

#### Topic 3: HTML: More Elements and Attributes

- Working with Text, Links, and Images
- Lists and Tables in HTML
- Semantic HTML: Using Headers, Paragraphs, and Semantic Tags

#### Topic 4: Introduction to CSS (Cascading Style Sheets)

- Basics of CSS: Selectors, Properties, Values
- Styling Text, Colors, and Backgrounds
- Applying CSS to HTML Documents

#### Topic 5: CSS Layout Techniques

- Box Model: Understanding Margin, Border, Padding
- Building Responsive Layouts with CSS
- Flexbox and Grid Layouts

#### Topic 6: Working with Forms and Input Elements

- Creating HTML Forms
- Form Controls: Textboxes, Radio Buttons, Checkboxes, Dropdowns

- Form Validation with HTML Attributes

#### Topic 7: Introduction to JavaScript

- Basics of JavaScript: Variables, Data Types, Operators
- Using JavaScript for Interactivity
- Introduction to Browser Console and Debugging Tools

#### Topic 8: DOM Manipulation with JavaScript

- Understanding the Document Object Model (DOM)
- Accessing and Modifying HTML Elements with JavaScript
- Handling Events: Click, Mouseover, Submit

#### Topic 9: Introduction to Responsive Web Design

- Importance of Responsive Design
- Using Media Queries for Responsive Layouts
- Introduction to Mobile-First Design Principles

#### Topic 10: Introduction to Web Development Tools

- Overview of Integrated Development Environments (IDEs)
- Introduction to Version Control with Git and GitHub
- Using Browser Developer Tools for Web Debugging

#### Topic 11: Introduction to Web Hosting and Deployment

- Basics of Web Hosting Services
- Deploying a Simple Website to a Web Hosting Provider
- Domain Names and DNS (Domain Name System)

#### Topic 12: Review and Project Showcase

- Recap of Key Concepts Covered
- Showcasing Projects Built During the Course

#### Assessment Criteria:

##### Quizzes (30%):

Quizzes will cover topics such as HTML structure, CSS styling, JavaScript basics, and responsive design principles. Quizzes will assess students' understanding of key concepts and their ability to apply web development techniques.

##### Mini Project (40%):

The Mini project will assess students' ability to apply their knowledge and skills to create a cohesive and visually appealing website that meets specified requirements.

##### Final Examination (30%):

The final examination will evaluate students' overall understanding of web development concepts, problem-solving abilities, and proficiency in utilizing web development tools.

#### Recommended Reading:

- “HTML and CSS: Design and Build Websites” by Jon Duckett, ISBN: 9781118008188
- “JavaScript and JQuery: Interactive Front-End Web Development” by Jon Duckett
- “Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics” by Jennifer Niederst Robbins, ISBN: 9781449319274

<b>Course Code</b>	<b>DIT12074</b>			<b>Course Title</b>	<b>Introduction to Programming</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>2</b>	<b>Credit</b>	<b>04</b>	<b>Theory (hr)</b>	30
						<b>Practical (hr)</b>	60
						<b>Independent Learning (hr)</b>	110
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to introduce students to the fundamental concepts of programming using Python, covering basic syntax, data types, control structures, and essential programming techniques.

### Intended Learning Outcomes:

- LO1: Describe the basic syntax and data types in Python.
- LO2: Apply control structures and functions to solve simple programming problems.
- LO3: Write Python programs to implement basic algorithms and solve simple computational tasks.

### Content:

#### Topic 1: Introduction to Python Programming

- Overview of Python
- Setting up Python Environment
- Writing and Executing Python Code

#### Topic 2: Basic Syntax and Data Types

- Variables and Data Types
- Operators and Expressions
- Input and Output Operations

#### Topic 3: Control Structures

- Conditional Statements (if, elif, else)
- Looping Constructs (for loop, while loop)
- Control Statements (break, continue)

#### Topic 4: Functions

- Defining and Calling Functions
- Function Parameters and Return Values
- Scope and Lifetime of Variables

#### Topic 5: Lists and Tuples

- Creating and Accessing Lists
- List Operations and Methods
- Working with Tuples

#### Topic 6: Dictionaries and Sets

- Creating and Accessing Dictionaries
- Dictionary Methods and Operations
- Working with Sets

#### Topic 7: File Handling

- Opening and Closing Files
- Reading and Writing Text Files

- File Handling Operations

#### Topic 8: Exception Handling

- Handling Exceptions with try-except blocks
- Raising Exceptions
- Finally block and Exception Hierarchy

#### Topic 9: Working with Modules and Packages

- Importing Modules
- Creating and Using Custom Modules
- Introduction to Packages

#### Topic 10: Basic Input Validation

- Validating User Input
- Handling Invalid Inputs
- Error Messages and Prompts

#### Topic 11: String Manipulation

- String Operations and Methods
- Formatting Strings
- Regular Expressions

#### Topic 12: Debugging and Troubleshooting

- Identifying and Fixing Common Errors
- Debugging Techniques and Tools
- Best Practices for Troubleshooting Python Code

#### Assessment Criteria:

Programming Assignments (60%):

Assignments requiring students to write Python programs to solve specific problems.

Final Examination (40%):

Comprehensive examination covering all topics, including theoretical concepts and practical programming tasks.

#### Recommended Reading:

- “Python Crash Course” by Eric Matthes, ISBN: 9781593277390
- “Automate the Boring Stuff with Python” by Al Sweigart
- “Introduction to Computation and Programming Using Python” by John V. Guttag, ISBN: 9780262529624



<b>Course Code</b>	<b>DIT12083</b>			<b>Course Title</b>	<b>Object Oriented Concepts</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>2</b>	<b>Credit</b>	<b>03</b>	<b>Theory (hr)</b>	20
						<b>Practical (hr)</b>	40
						<b>Independent Learning (hr)</b>	90
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to provide students with a foundational understanding of object-oriented programming (OOP) principles using C# / Java / Python, enabling them to design and implement efficient, scalable, and maintainable software solutions.

### Intended Learning Outcomes:

LO1: Demonstrate an understanding of the fundamental concepts of object-oriented programming, including classes, objects, inheritance, encapsulation, and polymorphism.  
LO2: Apply object-oriented programming principles to design and implement Python programs, creating classes, defining methods, and utilizing inheritance and polymorphism to build modular and reusable code.

### Content:

#### Topic 1: Introduction to Object-Oriented Programming (OOP)

- What is OOP and why is it important?
- Basic concepts: Classes, objects, attributes, methods
- Introduction to encapsulation and abstraction

#### Topic 2: Defining Classes and Creating Objects

- Creating classes and objects in Python
- Class attributes and instance variables
- Using methods to define behaviors

#### Topic 3: Constructors and Destructors

- Understanding the ‘`__init__()`’ method for initialization
- Using the ‘`__del__()`’ method for cleanup
- Constructor overloading and default arguments

#### Topic 4: Inheritance and Polymorphism

- Understanding inheritance and its benefits
- Creating subclasses and superclasses
- Method overriding and polymorphism in Python

#### Topic 5: More on Inheritance

- Multiple inheritance and method resolution order (MRO)
- Abstract base classes (ABCs) and interfaces
- Using ‘`super()`’ to call superclass methods

#### Topic 6: Encapsulation and Information Hiding

- Encapsulation and access control using private and public attributes
- Getter and setter methods for controlled access
- Property decorators for attribute access

Topic 7: Composition and Aggregation

- Understanding composition and aggregation
- Implementing composition relationships in classes
- Using aggregation to represent part-whole relationships

Topic 8: Class and Static Methods

- Defining class methods with '@classmethod'
- Creating static methods with '@staticmethod'
- Difference between instance, class, and static methods

Topic 9: Operator Overloading

- Introduction to operator overloading
- Implementing custom behavior for operators using magic methods
- Examples of common magic methods like `__add__()` and `__str__()`

Topic 10: Polymorphism and Duck Typing

- Understanding polymorphism and duck typing in Python
- Leveraging polymorphism to write flexible and reusable code
- Using interfaces and abstract classes for polymorphic behavior

Topic 11: Introduction to Design Patterns

- Overview of common design patterns in OOP
- Examples of creational, structural, and behavioral design patterns
- How design patterns promote reusable and maintainable code

Topic 12: Review, Project Showcase, and Conclusion

- Recap of key concepts covered throughout the course
- Showcase of projects demonstrating OOP principles

**Assessment Criteria:**

LMS based assessment (50%):

These will assess students' understanding of key concepts and their ability to apply object-oriented programming principles in C# / Java / Python.

Mini Project (50%):

The mini project will assess students' ability to apply object-oriented programming concepts to solve real-world problems, demonstrating their understanding of OOP principles and their practical application.

**Recommended Reading:**

- "Beginning C# Object-Oriented Programming" by Dan Clark, Jeff Sanders, ISBN: 9781430235316
- "Head First C#" by Andrew Stellman, Jennifer Greene, ISBN: 9781491976678
- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

<b>Course Code</b>	<b>DIT12093</b>			<b>Course Title</b>	<b>Networking and Security Essentials</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>2</b>	<b>Credit</b>	<b>03</b>	<b>Theory (hr)</b>	30
						<b>Practical (hr)</b>	30
						<b>Independent Learning (hr)</b>	90
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to provide students with a comprehensive understanding of computer networks, network protocols, and network security principles, enabling them to design, implement, and secure basic network infrastructures.

### Intended Learning Outcomes:

LO1: Demonstrate an understanding of fundamental concepts in computer networking, including network types, components, protocols, and models, as well as basic principles of network security.  
LO2: Apply their knowledge of networking concepts to configure network settings, troubleshoot common network issues, and implement basic security measures to protect network infrastructure.  
LO3: Analyze network security threats and vulnerabilities, evaluate security best practices, and develop basic security policies to mitigate risks and enhance network security.

### Content:

#### Topic 1: Introduction to Computer Networks

- What is a Computer Network?
- Types of Networks: LAN, WAN, WLAN
- Basic Network Components: Nodes, Routers, Switches

#### Topic 2: Understanding IP Addresses and Subnetting

- Introduction to IP Addresses: IPv4 vs. IPv6
- Basics of Subnetting: Network, Host, Subnet Mask
- Configuring IP Addresses on Devices

#### Topic 3: Introduction to Network Protocols

- Common Network Protocols: HTTP, FTP, DNS, DHCP
- Overview of OSI and TCP/IP Models
- Basics of Packet Switching and Data Transmission

#### Topic 4: Basics of Ethernet and Wireless Networks

- Understanding Ethernet and Wireless Technologies
- Introduction to Wi-Fi Standards (e.g., 802.11)
- Connecting Devices to a Wireless Network

#### Topic 5: Introduction to Network Security

- Importance of Network Security
- Common Network Threats: Malware, Phishing, Denial of Service (DoS)
- Basic Security Practices: Passwords, Firewalls, Updates

#### Topic 6: Introduction to Encryption and Cryptography

- Basics of Encryption and Decryption
- Types of Cryptographic Algorithms: Symmetric and Asymmetric

- Using HTTPS for Secure Communication

Topic 7: Introduction to Firewalls and Antivirus Software

- Basics of Firewalls: Types and Functionality
- Using Antivirus Software for Threat Detection
- Configuring Firewall Rules for Network Security

Topic 8: Wireless Network Security

- Securing Wi-Fi Networks: WPA/WPA2, SSID Broadcasting
- Basic Wireless Security Settings on Routers
- Identifying and Preventing Unauthorized Access

Topic 9: Introduction to VPNs (Virtual Private Networks)

- Understanding VPN Concepts and Benefits
- Types of VPNs: Site-to-Site vs. Remote Access
- Using VPN Clients for Secure Connections

Topic 10: Network Monitoring and Troubleshooting

- Basics of Network Monitoring Tools (e.g., Ping, Traceroute)
- Identifying Common Network Issues
- Troubleshooting Connectivity Problems

Topic 11: Security Best Practices and Awareness

- Implementing Basic Security Policies
- Importance of Security Awareness Training
- Recognizing Social Engineering Attacks

Topic 12: Review and Practical Applications

- Recap of Key Concepts Covered
- Hands-On Exercises: Configuring Network Settings, Basic Security Configurations

**Assessment Criteria:**

LMS based assessment (60%):

This will cover topics such as network types, components, protocols, security threats, and security best practices. Regular quizzes will assess students' understanding of key networking and security concepts and their ability to apply this knowledge in various scenarios.

Final Examination (40%):

The final examination will comprehensively cover all topics taught throughout the course, including network types, protocols, security principles, and practical applications.

**Recommended Reading:**

- “Computer Networking: A Top-Down Approach” by James F. Kurose and Keith W. Ross, ISBN: 9781292153605
- “Network Security Essentials: Applications and Standards” by William Stallings
- “Cisco Networking All-in-One for Dummies” by Edward Tetz, ISBN: 9781118137857

<b>Course Code</b>	<b>DIT12105</b>			<b>Course Title</b>	<b>IT Project Development</b>		
<b>Level</b>	<b>3</b>	<b>Semester</b>	<b>2</b>	<b>Credit</b>	<b>05</b>	<b>Theory (hr)</b>	20
						<b>Practical (hr)</b>	-
						<b>Independent Learning (hr)</b>	230
<b>Status</b>	Core Mandatory			<b>Pre-requisite Course/s</b>	-		

### Aim of the Course:

This course aims to provide students with a foundational understanding of IT project development processes, including project planning, implementation, testing, documentation, and presentation, to equip them with the skills needed to undertake basic software development projects.

### Intended Learning Outcomes:

LO1: Demonstrate an understanding of fundamental concepts in IT project development, including project planning, programming basics, testing methodologies, and project presentation skills.  
LO2: Apply their knowledge of IT project development concepts to plan, implement, and document basic software projects, incorporating user feedback and conducting testing and debugging activities.  
LO3: Evaluate project progress and outcomes through peer review and feedback, refining project features and presentation skills based on constructive criticism.

### Content:

#### Topic 1: Introduction to IT Project Development

- Overview of the module objectives and project requirements.
- Introduction to project ideation and brainstorming.

#### Topic 2: Requirement Gathering and Planning

- Defining project scope and objectives.
- Conducting user interviews and gathering requirements.

#### Topic 3: Basic Programming Concepts

- Introduction to programming languages (e.g., Python).
- Basics of variables, data types, and control structures.

#### Topic 4: Designing the Project

- Creating project wireframes or mockups.
- Understanding user interface (UI) design principles.

#### Topic 5: Implementing the Project (Part 1)

- Setting up development environments.
- Writing basic code snippets for project functionalities.

#### Topic 6: Implementing the Project (Part 2)

- Incorporating user feedback and iterative development.
- Collaborative coding and version control using Git.

#### Topic 7: Testing and Debugging

- Basics of software testing techniques (e.g., unit testing).
- Identifying and fixing common bugs and errors.

Topic 8: Project Documentation

- Writing technical documentation (e.g., README file, code comments).
- Documenting project functionalities and implementation details.

Topic 9: Project Presentation Skills

- Basics of effective project presentation.
- Creating engaging project demos and presentations.

Topic 10: Peer Review and Feedback

- Peer evaluation of project progress and outcomes.
- Providing constructive feedback for improvement.

Topic 11: Finalizing and Polishing the Project

- Conducting final testing and quality assurance.
- Refining project features and addressing last-minute issues.

Topic 12: Project Showcase and Evaluation

- Showcase of student projects to peers and instructors.
- Evaluation based on project completeness, functionality, and presentation.

**Assessment Criteria:**

Project proposal and presentation (20%):

Students are required to create a software proposal, present it, and obtain approval.

Project Presentation and Peer Review (50%):

Project presentation and peer review will involve showcasing student projects to peers and instructors, as well as providing and receiving constructive feedback for improvement. Students will be evaluated based on the completeness, functionality, and presentation of their projects, as well as their ability to incorporate feedback and iterate on project features.

Final project report (30%)

Students are required to submit the final hardcopy of the project report, addressing the reviewer's comments

**Recommended Reading:**

- “The Art of Agile Development” by James Shore and Shane Warden, ISBN: 9780596527679
- “Clean Code: A Handbook of Agile Software Craftsmanship” by Robert C. Martin
- “Agile Estimating and Planning” by Mike Cohn, ISBN: 9780131479418