



## Scope and Sequence for Information Technology 2

---

### Course Description

This dynamic and immersive course builds upon foundational IT skills, guiding students through an exciting exploration of intermediate and advanced concepts across a diverse range of cutting-edge technologies. Students will delve deeper into the intricate workings of operating systems, harness the power of command-line interfaces, craft dynamic web experiences, and refine their programming expertise. They will push the boundaries of creativity with 3D modeling, design fully immersive virtual environments, experiment with robotics, and develop interactive applications that bring their ideas to life.

Throughout the course, students will engage in hands-on projects that not only reinforce critical problem-solving skills but also enhance logical thinking, computational reasoning, and design methodologies. By working with parametric modeling, real-time interactivity, automation, and virtual simulations, students will gain a well-rounded technical foundation that prepares them for advanced studies and real-world innovation. This course is designed to inspire curiosity, encourage experimentation, and equip students with the tools to shape the future of technology.

---

### Module 1: Operating Systems and Command Line Interfaces

This module introduces students to the command-line interface (CLI), an essential tool for managing files, executing programs, and customizing user environments.

#### Key Topics:

- Overview of operating systems, their structure, and their role in computing.
- Introduction to the command line: how it differs from graphical user interfaces.
- Navigating the file system: working with absolute and relative paths.

- Common commands for file and directory management:
  - pwd (print working directory)
  - ls (list files)
  - cd (change directory)
  - mkdir and rmdir (create and remove directories)
  - touch, rm, cp, mv (file management).
- File permissions and ownership: understanding read, write, and execute permissions.
- Environment customization: configuring the shell using startup scripts.
- Basic text editing in CLI-based editors and the importance of version control.

### Skills Developed:

- Efficiently navigating a Unix-like file system using CLI commands.
- Managing files and directories programmatically.
- Understanding file permissions and modifying them appropriately.
- Customizing the user environment through profile scripts and aliases.

### Assignments/Projects:

- Navigate and manipulate directories through CLI commands.
  - Customize the shell prompt and define aliases for frequently used commands.
  - Use a command-line text editor to modify and organize files.
- 

## Module 2: Web Design and Development

Students learn the structure of web pages, incorporating styling and layout techniques for responsive design.

### Key Topics:

- Introduction to markup and styling languages for structuring and designing web pages.
- Semantic elements for better accessibility and organization of content.
- Styling techniques, including cascading styles, fonts, colors, and layout positioning.
- Responsive design principles:
  - Implementing flexible grids and layouts.
  - Using media queries to adapt to different screen sizes.
- Best practices for web accessibility and user experience.

### Skills Developed:

- Structuring web pages with semantic and well-organized markup.
- Applying styles to create visually appealing and responsive designs.
- Understanding how web pages adapt to various devices.

## Assignments/Projects:

- Develop a webpage containing text, images, links, and styled sections.
  - Create a responsive design layout that adjusts to different screen sizes.
  - Implement accessibility features such as alternative text for images.
- 

## Module 3: Interactive Programming for Web Applications

This module introduces students to programming concepts that enhance interactivity in web pages.

### Key Topics:

- Programming fundamentals: variables, loops, conditional statements.
- Manipulating the Document Object Model (DOM) dynamically.
- Handling user events such as clicks, form submissions, and keypresses.
- Writing functions to handle repetitive tasks efficiently.
- Debugging techniques and error handling.
- Asynchronous programming: making requests to external data sources.

### Skills Developed:

- Writing and executing scripts for interactivity.
- Manipulating elements on a web page dynamically.
- Handling user input and responding to interactions.

## Assignments/Projects:

- Create an interactive feature that responds to user input.
  - Develop a browser-based application that updates content dynamically.
  - Implement a small game or visual effect using programming logic.
- 

## Module 4: 3D Modeling and Parametric Design

Students explore 3D modeling concepts, using scripting to create objects for digital use or 3D printing.

### Key Topics:

- Primitive 3D shapes and their properties.
- Boolean operations: union, difference, intersection.
- Transformations: translation, rotation, and scaling.
- Parametric design: using variables to define customizable models.
- Adding detail and smoothing surfaces with geometric modifiers.

- Preparing models for 3D printing: exporting and checking dimensions.

### Skills Developed:

- Creating 3D objects through scripting and procedural design.
- Using parametric modeling to allow for customization.
- Understanding best practices for printable models.

### Assignments/Projects:

- Design a functional 3D object such as a modular container.
  - Create an optimized and customizable 3D model with adjustable parameters.
  - Export a model for 3D printing.
- 

## Module 5: Virtual and Immersive Environments

Students build interactive virtual spaces and explore concepts of realism and physics.

### Key Topics:

- Constructing environments with geometric shapes and textured surfaces.
- Adjusting lighting to enhance realism.
- Animating objects within a scene.
- Adding physics-based behaviors such as collisions and movement constraints.
- Interactive components: user-triggered events and scripted actions.

### Skills Developed:

- Creating immersive scenes with realistic lighting and textures.
- Designing interactions within a virtual environment.
- Using physics-based interactions for more natural movement.

### Assignments/Projects:

- Develop a virtual environment with customized textures and animations.
  - Implement user-triggered events such as interactive doors or buttons.
  - Enhance realism by integrating dynamic lighting and object physics.
- 

## Module 6: Robotics Programming and Automation

Students program robotic behavior and explore automated movement algorithms.

### Key Topics:

- Programming motion control with motor functions.
- Implementing sensor-based navigation for obstacle avoidance.
- Developing pathfinding algorithms.
- Using loops and conditions for autonomous behavior.

### Skills Developed:

- Writing scripts for robotic movement and control.
- Implementing logic for autonomous decision-making.
- Debugging and optimizing robot motion paths.

### Assignments/Projects:

- Program a robot to complete an obstacle course.
  - Implement a maze-solving algorithm using real-time sensor input.
  - Create an interactive program that modifies robotic movement based on user commands.
- 

## Module 7: Interactive Application Development

Students apply their knowledge to design interactive applications.

### Key Topics:

- Implementing input-based controls for navigation.
- Adding scoring and tracking progress.
- Integrating animations and physics into interactions.
- Structuring applications using modular programming techniques.

### Skills Developed:

- Developing applications with real-time interaction.
- Managing game state and user progress.
- Integrating physics and animation into digital experiences.

### Assignments/Projects:

- Design an application that tracks user actions.
  - Implement a system for responding to user input dynamically.
  - Develop a simple interactive project with animation and effects.
-

## Module 8: Capstone Project

Students integrate knowledge from all modules into a final project.

### Key Topics:

- Planning and designing a full-scale project.
- Combining multiple technical disciplines into a cohesive experience.
- Documenting and presenting project development.

### Skills Developed:

- Structuring and executing a self-directed project.
- Combining multiple technologies into a single application.
- Presenting and explaining a technical project.

### Assignments/Projects:

- Develop a comprehensive project incorporating skills from previous modules.
- Write a project summary detailing the development process.
- Present and showcase the final project.

**\*Disclaimer:** The course structure and content outlined in this scope and sequence are subject to change. MYTEK LAB reserves the right to adjust the order, pacing, and topics covered to best meet the needs of students and ensure an optimal learning experience.