



Robotics for Professional

Week 1 Introduction to Linux

Session 1 Objective: Introduce students to the Linux operating system and its core concepts.

- What is Linux?
- History of Linux and its distributions
- Linux as an open-source operating system
- Introduction to the Linux terminal
- Basic terminal commands (ls, cd, pwd, mkdir, rmdir)
- File and directory structure in Linux
- Navigating the file system

Session 2: Users and Permissions

- Objective: Teach users and permissions management in Linux.
- User accounts in Linux
- Creating and deleting user accounts
- User groups and group management
- File and directory permissions (chmod, chown)
- Understanding permission modes (rwx)
- Changing ownership and permissions
- Managing files and directories as a regular user and superuser (sudo)

Week 2: File System and operations

Session 3: File Operations and Processes

Objective: Cover file operations, text processing, and managing processes in Linux.

- Working with files (touch, cp, mv, rm)
- File archiving and compression (tar, gzip)
- Text editors in Linux (nano, vim)
- Basic text manipulation (grep, cat, less)
- Processes and process management (ps, top, kill)
- Running and managing background processes
- Redirecting input and output (>, >>, <, |)

Session 4: System Management and Basic Scripting

- Objective: Explore system management tasks and introduce basic scripting concepts.
- System information commands (uname, df, du)
- Managing services and daemons (systemctl)
- Package management (apt or yum)
- Basic shell scripting concepts
- Writing and running simple shell scripts
- Script execution permissions
- Automating repetitive tasks with scripts
- Examples of common scripting tasks (e.g., file backups)

Week 3: Basics of CPP Language

Introduction to Programming using C++

- Data Types.
- Variables.
- Operators.
- Control Flow and repetitions.
- Namespaces

week 4: Modular programming in practice

- Preprocessor
- Functions
- Pointers (pointer to every data type, pointer to functions).
- Lambda expression
- Callback.
- Reference.
- Arrays and vectors.

Week 5: Object Oriented Programming

- Object Oriented Programming
- Classes and Objects
- Advanced class features
- Inheritance
- Compile time Polymorphism
- Run-time Polymorphism and abstraction

Week 6: OOP Relationships and Templates

- UML and design concepts
- OOP Relationships
- Composite Pattern
- C++ Templates
- Strategy design pattern

Week 7: STL (Containers and Algorithms)

- Big O notation
- Linear Data structure (array, vector, linked list, stack and queue)
- Associative Containers (set, map, unordered set and unordered map)

Week 8: Smart Pointers

- Raw Pointers problems
- Unique pointer, shared pointer and weak pointer
- Cyclic dependency problem and it's solution
- Dependency Injection pattern

Week 9: SOLID and design patterns

- SOLID principles
- Design patterns (Observer and Factory)

Week 10: Continue Patterns and Complete Build Tools

- Design patterns (Proxy and Adaptor)
- Makefiles scripts and their benefits
- CMake for medium and large sized projects

Week 11: SW Engineering

- Requirements analysis
- SDLC, Agile and V-Model
- Unit Testing using GoogleTest framework
- Test driven development

Week 12: Git and DevOps

- Version control using git
- Rebasing and merging
- Learn about dockers
- Create a pipeline using jenkins