



Arduino IDE + Thonny — Complete Course Content

WC Skills • World Class Skills Institute

Duration	6 Weeks
Target Audience	Learners combining embedded C/C++ with Python/MicroPython
Prerequisites	Basic computer skills.

What this document contains

A module-wise training roadmap covering concepts, hands-on labs, project work, assessment checkpoints and interview preparation. The syllabus is designed as a structured course plan for classroom or live-online delivery.

Course Objectives & Learning Outcomes

By the end of this course, learners should be able to explain the core concepts, perform the practical tasks listed in the labs, troubleshoot common scenarios, document their work and complete the capstone with a professional implementation approach.

- Understand the architecture, terminology and end-to-end workflow of the subject.
- Perform guided hands-on exercises using the relevant tools and development/administration environment.
- Apply troubleshooting, testing, documentation and best-practice techniques.
- Build a portfolio-ready capstone demonstrating practical competency.
- Prepare for technical interviews with scenario-based questions and implementation discussions.

Detailed Module-wise Curriculum

Module 1: Toolchain & Boards

- Arduino IDE setup
- Thonny installation
- Board drivers
- Serial ports
- Project organization

Hands-on Lab: Lab: prepare both toolchains and verify serial connectivity.

Module 2: Arduino Programming

- Sketch structure
- GPIO
- PWM
- analog input
- Libraries

Hands-on Lab: Lab: build a sensor-controlled actuator.

Module 3: Python & Thonny

- Python syntax
- Variables
- functions
- collections
- debugger

Hands-on Lab: Lab: write a serial-data monitor in Python.

Module 4: MicroPython & Microcontrollers

- MicroPython concepts
- REPL
- Thonny device connection
- GPIO
- Sensor reading

Hands-on Lab: Lab: run a MicroPython sensor program.

Module 5: PC-to-Board Communication

- Serial protocol design
- Data formats
- Command/response
- Logging
- Error handling

Hands-on Lab: Lab: control a board from a Python desktop script.

Module 6: Capstone

- System design
- Testing
- Troubleshooting
- Documentation
- Demo

Hands-on Lab: Lab: deliver a Python + microcontroller automation solution.

Practical Delivery Plan

Phase	Focus	Expected Evidence
1	Concept + demonstration	Notes, diagrams and guided exercises
2	Hands-on practice	Working configuration/code and screenshots/logs
3	Troubleshooting	Issue reproduction, diagnosis and resolution notes
4	Mini project	Functional solution with test cases
5	Capstone	End-to-end project, documentation and presentation

Assessment & Certification Readiness

Recommended assessment pattern: module quizzes (20%), practical labs (30%), mini project (20%), capstone (20%) and interview/viva (10%). Completion certificates, placement support and any certification claims should follow the institute's current published policy and the relevant vendor's certification rules.

Capstone Requirements

- Problem statement and scope
- Architecture/process diagram
- Implementation or configuration
- Test cases and results
- Troubleshooting notes
- User/technical documentation
- Final presentation and interview questions

Interview Preparation Topics

Learners should be able to answer architecture questions, explain the difference between key concepts, troubleshoot a real scenario, justify design choices, discuss performance/security considerations and walk through the capstone from requirement to deployment/support.

Recommended Learning Resources

Use official product documentation and current vendor guidance for version-specific commands, configuration steps and certification requirements. Validate training examples in a non-production environment before applying them to customer systems.