Faysal Aziz

Raspberry Pi Mastery

Python, C++, and Java for Embedded Development - 3 days

FA-0396 · 3 days

Software Development · DevOps, Cloud & Infrastructure · Automation, RPA & Power Platform

---

Introduction

In the rapidly evolving landscape of embedded systems and Internet of Things (IoT), the Raspberry Pi stands out as a versatile and accessible platform for developers and engineers. This course is meticulously designed to provide participants with hands-on experience in configuring embedded Linux on Raspberry Pi and developing applications using Python, C++, and Java.

Through real-world projects, attendees will gain the skills necessary to interface with hardware components, set up web and FTP servers, create web-based interfaces, develop network applications, integrate SQL databases, and automate system tasks with shell scripting. Guided by an instructor with over 30 years of industry experience, this intensive 3-day course emphasizes practical, industry-relevant content over academic theory.

Learning outcomes

By the end of this course, participants will be able to:

- Set up and configure embedded Linux on Raspberry Pi.
- Develop applications using Python, C++, and Java on Raspberry Pi.
- Interface GPIO pins to control hardware components like LEDs and sensors.
- Set up and manage Apache web and FTP servers.
- Create web-based interfaces for remote device control.
- Develop network applications using TCP and UDP.
- Use SQL databases to store and retrieve sensor and system data.
- Automate system tasks with shell scripting.

Prerequisites

This course is designed for participants with a solid foundation in programming and system administration. To ensure a productive learning experience, attendees must meet the following prerequisites:

- **Programming Proficiency:** Demonstrated experience in Python, C++, and Java. Participants should be comfortable writing, debugging, and compiling code in these languages.
- **Linux Command-Line Experience:** Familiarity with Linux command-line operations, including file system navigation, package management, and basic shell scripting.
- **Hardware Interaction:** Basic understanding of electronic components and experience with hardware interfacing is highly recommended.
- **Raspberry Pi Setup:** Participants must bring a Raspberry Pi (Model 3 or later) with Raspberry Pi OS installed, along with necessary peripherals (power supply, SD card, monitor, keyboard, and mouse).
- **Software Installation:** Ability to install and configure development environments and necessary libraries on Raspberry Pi.

Note: This course is intensive and assumes a high level of technical proficiency. Participants who do not meet the prerequisites may find the material challenging.

Training outline

1. 1. Introduction to Raspberry Pi

- Overview of Raspberry Pi models and hardware architecture.
- Practical applications in IoT, automation, and edge computing.
- Setting up Raspberry Pi for development and real-world projects.

2. 2. System Setup and Linux Configuration

- Installing and configuring Raspberry Pi OS (Raspbian).
- Initial setup: user configurations, SSH access, and remote login.
- Managing packages using apt (Advanced Package Tool).
- Connecting Raspberry Pi to the internet and troubleshooting network issues.

3. 3. Setting Up the Development Environment

Python Development:

- Installing Python and essential libraries.
- Running Python scripts and managing dependencies.

C++ Development:

- Installing GCC and configuring a C++ development environment.
- Writing, compiling, and running simple C++ programs.

Java Development:

- Installing Java Development Kit (JDK) and NetBeans.
- Developing simple Java GUI applications using Swing and JavaFX.

4. 4. Interfacing with GPIO (General-Purpose Input/Output)

- Understanding the I/O architecture of Raspberry Pi.

  Python GPIO Programming:

  - Using RPi.GPIO and gpiozero libraries.
  - Controlling LEDs and reading sensor inputs.

  C++ GPIO Programming:

  - Using pigpio for hardware interaction citeturn0search0.
  - Implementing PWM for motor control.

  Java GPIO Programming:

  - Using Pi4J to interact with GPIO.
  - Creating Java-based applications that control electronic components.

5. 5. Web Server Deployment on Raspberry Pi

- Understanding daemons in Linux.

  Introduction to Apache Web Server:

  - Installation and configuration on Raspberry Pi citeturn0search7.
  - Managing server operations: starting, stopping, and troubleshooting.

  Creating a Python-Based Web Application:

  - Using Flask to develop a web interface for GPIO control.
  - Implementing a REST API to interact with Raspberry Pi hardware.

6. 6. FTP Server Setup

- Installing and configuring vsftpd on Raspberry Pi citeturn0search3.
- Managing FTP users and securing file access.
- Transferring data between Raspberry Pi and other devices.

7. 7. Web-Based Control and IoT Applications

- Designing and developing static and dynamic web pages.
- Enhancing interactivity using AJAX.

  Web-Based GPIO Control:

  - Controlling LEDs and reading sensor data via a browser.

© 2026 Faysal Aziz · Kuala Lumpur