

Raspberry Pi Mastery

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

FA-0397 · 3 days

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

Introduction

Unlock the Power of Raspberry Pi with Real-World Applications

Ready to unleash the power of Raspberry Pi? Join us for three action-packed days where you'll learn to build everything from IoT devices to automated systems. This isn't your typical theoretical course – you'll be learning from someone who's spent three decades in the trenches of tech development. Whether you're a developer, engineer, or tech enthusiast, you'll get hands-on experience with Python, C++, and Java to create real-world applications.

We'll cover it all: setting up embedded Linux, programming in multiple languages, connecting hardware through GPIO, networking, and creating web interfaces to control your projects. Best of all, you'll be working with the Raspberry Pi – an incredibly versatile and affordable platform that's perfect for bringing your ideas to life. By the time you're done, you'll have the skills to build your own automated systems, IoT devices, and remote-controlled applications.

Learning outcomes

- 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

- Basic programming knowledge (Python, C++, and Java is a must).
 - Familiarity with **Linux command-line** operations.
 - Basic understanding of **electronics and networking**.
-

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 on Raspberry Pi:

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

C++ development on Raspberry Pi:

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

Java development on Raspberry Pi:

- 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 **wiringPi** for hardware interaction.
- 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.
- 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.
- **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.
- Using Python-based web frameworks (**Flask/Django**) for dynamic interfaces.

8. 8. File Handling and Database Integration

- **Reading and writing files** using Python and Java.

Introduction to **SQL databases** on Raspberry Pi.

- Installing and using **SQLite/MySQL**.
- Executing SQL queries to store and retrieve sensor data.
- Developing **Python and Java applications** to interact with databases.

9. 9. Network Communication and Client-Server Applications

- Fundamentals of **TCP and UDP** communication protocols.

Developing **Python client-server applications**:

- Implementing TCP sockets for data exchange.
- Using UDP for low-latency communication.

Java networking:

- Creating Java-based client-server applications.
- Handling real-time data transmission.

10. 10. Shell Scripting and Automation

- Introduction to **shell scripting in Linux**.
- Writing **bash scripts** to automate system tasks.

Automating:

- **Software updates, backups, and network monitoring.**
 - **GPIO-based event triggers and alerts.**
-

Conclusion and Final Project

Participants will apply their skills in a **mini-project**, combining **Python, C++, Java, and web interfaces** to build a **functional Raspberry Pi-based application**.

Examples of possible projects:

- **Smart Home Automation System:** Control appliances using a web dashboard.
 - **Remote Surveillance System:** Capture images with a Raspberry Pi camera and store them in a database.
 - **Networked Sensor System:** Collect sensor data and transmit it over a network.
-

By the end of this course, participants will have a **strong foundation in Raspberry Pi programming and system development** using Python, C++, and Java. With hands-on training and real-world applications, they will be well-equipped to develop **IoT solutions, embedded systems, and automation projects**.