



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
PRC-39

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

PROGRAMMING / RUST

Rust Fundamentals with Generative AI

เขียน Rust อย่างเข้าใจกลไก แล้วสร้าง Service ที่เชื่อมต่อ Generative AI

- 30 ชั่วโมง
- 5 วัน
- พื้นฐาน เขียนโปรแกรมได้อย่างน้อย 1 ภาษา
- ระดับ Beginner to Intermediate



ระยะเวลา: 5 วัน

30 ชั่วโมง (5 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

องค์กรจำนวนมากเริ่มนำ Generative AI เข้าไปอยู่ในระบบงานจริง ทั้งการสรุปเอกสาร การจัดหมวดหมู่คำร้อง และผู้ช่วยตอบคำถาม ระบบเหล่านี้ต้องทำงานเร็ว ใช้ทรัพยากรน้อย และเสถียรเมื่อมีผู้ใช้จำนวนมาก Rust จึงถูกเลือกใช้มากขึ้นสำหรับงาน Backend และ Service ที่ต้องการความปลอดภัยของหน่วยความจำและประสิทธิภาพสูง แต่สำหรับนักพัฒนาที่มาจากภาษาอื่น แนวคิดอย่าง Ownership, Borrowing และ Lifetimes มักเป็นกำแพงแรกที่ทำให้เรียนรู้ได้ช้า

หลักสูตรนี้ปูพื้นฐานภาษา Rust อย่างเป็นลำดับตั้งแต่ Toolchain ไปจนถึงกลไกสำคัญของภาษา แล้วต่อยอดสู่การเขียนโปรแกรมแบบ Asynchronous ด้วย Tokio สร้าง Web API ด้วย axum และเชื่อมต่อ LLM ทั้งแบบ Cloud ผ่าน OpenAI Responses API และแบบ Local ผ่าน Ollama ทุกวันต่อยอดโปรเจกต์ TicketPilot ชัดเดียว จนได้ Service ที่ใช้ Generative AI วิเคราะห์ Ticket จำลองและพร้อมรับใน Docker Container ในวันสุดท้าย (อบรม 5 วัน วันละ 6 ชั่วโมง รวม 30 ชั่วโมง ระดับ Beginner to Intermediate)

เครื่องมือ

Rs Rust

Tk Tokio & axum

GP OpenAI API

OI Ollama

AI Generative AI

ราคาค่าอบรม · PUBLIC / IN-HOUSE

11,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide รายวัน โค้ดตัวอย่าง ข้อมูลจำลองสำหรับ Workshop และ Certificate · ค่าใช้งาน OpenAI API ผู้เรียนรับผิดชอบเอง · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- ติดตั้งและใช้งาน Rust Toolchain (rustup, cargo, clippy, rustfmt) และจัดโครงสร้างโปรเจกต์ได้อย่างถูกต้อง
- เข้าใจไวยากรณ์ ชนิดข้อมูล Struct, Enum และ Pattern Matching ของภาษา Rust
- เข้าใจและประยุกต์ใช้ Ownership, Borrowing และ Lifetimes พร้อมอ่าน Error ของ Borrow Checker ได้อย่างมีเหตุผล
- ออกแบบโค้ดด้วย Traits และ Generics และจัดการ Error ด้วย Result อย่างเป็นระบบ
- เขียนโปรแกรม Asynchronous ด้วย Tokio และสร้าง Web API ด้วย axum ได้
- เชื่อมต่อ LLM ผ่าน OpenAI Responses API และ Ollama จากโค้ด Rust ได้
- รับผลลัพธ์จาก LLM แบบ Structured Output แล้วแปลงเป็นชนิดข้อมูล Rust ด้วย serde อย่างปลอดภัย
- เขียน Unit Test และนำ Service ขึ้นรันใน Docker Container ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend ที่ต้องการเพิ่ม Rust สำหรับงานที่ต้องการประสิทธิภาพสูง
- นักพัฒนาที่ต้องการสร้าง Service หรือเครื่องมือภายในที่เชื่อมต่อกับ LLM
- ทีมพัฒนาที่กำลังพิจารณานำ Rust มาใช้ในระบบขององค์กร
- Software Engineer ที่ต้องการเข้าใจการจัดการหน่วยความจำแบบไม่ใช้ Garbage Collector
- ผู้ที่ผ่านหลักสูตร Rust Programming แล้วและต้องการฝึกต่อเนื่องจนสร้าง Service ที่ใช้ AI ได้

03 จุดเด่นของหลักสูตร Highlights

- เนื้อหาตรงกับ Rust ปัจจุบัน Rust 1.99, Edition 2024 และ axum 0.8
- ปูพื้นฐานไวยากรณ์เป็นลำดับ Ownership และ Borrow Checker แบบเข้าใจเหตุผล
- เชื่อม AI ได้ทั้ง Cloud และ Local OpenAI Responses API และ Ollama
- โปรเจกต์เดี่ยวต่อเนื่อง 5 วัน TicketPilot จาก CLI สู่ AI Service
- แนวปฏิบัติสำหรับงานจริง Error Handling, tracing, Test และ Docker

04 ความรู้พื้นฐาน Prerequisites

- เขียนโปรแกรมได้อย่างน้อย 1 ภาษา เช่น Python, JavaScript/TypeScript, Java, C# หรือ Go
- เข้าใจแนวคิดพื้นฐาน เช่น ตัวแปร ฟังก์ชัน เงื่อนไข ลูป และโครงสร้างข้อมูล
- ใช้งาน Command Line และ Git เบื้องต้นได้ และเข้าใจพื้นฐาน HTTP และ JSON
- ไม่จำเป็นต้องมีประสบการณ์ Rust, C หรือ C++ และไม่ต้องมีพื้นฐาน Machine Learning มาก่อน

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
ภาษาและ Toolchain	Rust 1.99 (stable), Edition 2024, rustup, cargo, clippy, rustfmt และ VS Code กับ rust-analyzer
Async และ Web API	Tokio, axum 0.8, tower, reqwest, serde, thiserror, anyhow, clap และ tracing
Generative AI	OpenAI Responses API ผ่าน crate async-openai 0.42 (โมเดลตระกูล GPT-6) และ Ollama กับโมเดล Open-weight ขนาดเล็ก
Container	Docker Desktop (Multi-stage Build) และ Docker Compose สำหรับรันคู่กับ Ollama
อุปกรณ์ผู้เรียน	Notebook Windows หรือ macOS RAM 16 GB ขึ้นไป พื้นที่ว่าง 30 GB และสักร์ติดตั้งโปรแกรม

หมายเหตุ: หลักสูตรนี้ต่อยอดจากหลักสูตร Rust Programming (2 วัน) ที่เน้นพื้นฐานภาษา โดยใช้เวลา 5 วันฝึกกลไกภาษาให้แน่นแล้วนำไปสร้าง Async Web Service ที่เชื่อมต่อ Generative AI ผู้เรียนใช้ OpenAI API Key ของตนเองหรือขององค์กร ซึ่งมีค่าใช้จ่ายตามการใช้งาน ผู้ที่ไม่มี Key ทำ Lab ฝั่ง AI ด้วย Ollama บนเครื่องตนเองได้ ผู้ใช้ Windows ควรติดตั้ง Visual Studio Build Tools และเปิด WSL2 สำหรับ Docker Desktop ส่วนหน้า เวอร์ชันของเครื่องมือและชื่อโมเดลเปลี่ยนบ่อย เนื้อหาจะอิงเวอร์ชันล่าสุดในวันอบรม

06 ภาพรวมการเรียนรู้ 5 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	ระบบนิเวศ Rust และ Edition 2024, ติดตั้ง rustup, cargo, clippy และ rustfmt, ไวยากรณ์และชนิดข้อมูล, Struct, Enum และ Pattern Matching แล้วเริ่ม TicketPilot แบบ CLI
Day 2 Section 6-10	Ownership, Move และ Drop, Borrowing และการอ่าน Error ของ Borrow Checker, Lifetimes เบื้องต้น, String, Slice และ Collections พร้อมลดการ clone ที่ไม่จำเป็น
Day 3 Section 11-15	Traits และ Generics, Closures และ Iterators, Error Handling ด้วย Result, thiserror และ anyhow, อ่านเขียน JSON ด้วย serde และ Unit Test ด้วย cargo test
Day 4 Section 16-20	Async Rust บน Tokio, Web API ด้วย axum, พื้นฐาน LLM สำหรับนักพัฒนา, เรียก OpenAI Responses API ด้วย async-openai และรัน Local LLM ด้วย Ollama
Day 5 Section 21-25	LLM Provider แบบสลับได้ด้วย Trait, Structured Output, Streaming ผ่าน SSE, Tool Calling เบื้องต้น, tracing, Config และ Docker แล้วรวมเป็น TicketPilot AI Service

DAY
1

Rust Foundations และ Toolchain

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 ทำความรู้จัก Rust และระบบนิเวศ

- เหตุผลที่องค์กรเลือก Rust: Memory Safety, ประสิทธิภาพ และความปลอดภัยของ Concurrency
- งานที่ Rust เหมาะและไม่เหมาะ เมื่อเทียบกับ Go, Java, C# และ Python
- รอบการออกเวอร์ชันทุก 6 สัปดาห์ แนวคิด Edition และสิ่งที่เปลี่ยนใน Edition 2024
- crates.io, docs.rs, The Rust Book และแหล่งเรียนรู้หลักของชุมชน

2 Lab: ติดตั้งและใช้งาน Toolchain

- ติดตั้ง rustup เลือก Toolchain แบบ stable และอัปเดตเวอร์ชัน
- คำสั่งหลักของ cargo: new, build, run, check, add และโครงสร้าง Cargo.toml
- จัดรูปแบบโค้ดด้วย rustfmt และตรวจสอบคุณภาพด้วย clippy
- ตั้งค่า VS Code กับ rust-analyzer และการอ่านข้อความ Error ของ Compiler
- Lab: สร้างโปรเจกต์แรกและแก้คำเตือนจาก clippy

3 ไวยากรณ์และชนิดข้อมูลพื้นฐาน

- ตัวแปร Immutability โดยค่าเริ่มต้น, mut และ Shadowing
- Scalar Types, Tuple, Array และการแปลงชนิดข้อมูลอย่างปลอดภัย
- ฟังก์ชัน Expression กับ Statement และการคืนค่า
- Control Flow: if, loop, while, for และ Range
- Lab: แบบฝึกหัดเขียนฟังก์ชันคำนวณและวนลูปข้อมูล

4 Struct, Enum และ Pattern Matching

- นิยาม Struct และ Method ด้วย impl รวมถึง Associated Function
- Enum ที่เก็บข้อมูลได้ และ Option ที่ใช้แทนค่า null
- match, if let และ let else สำหรับแยกกรณีอย่างครบถ้วน
- Lab: ออกแบบ Struct Ticket และ Enum สถานะกับระดับความเร่งด่วน

5 Workshop: Module และ TicketPilot CLI

- mod, pub, use และการแยกไฟล์ตามหน้าที่
- Library Crate กับ Binary Crate และการจัดโครงสร้างโปรเจกต์
- รับ Argument จาก Command Line ด้วย clap เบื้องต้น
- Workshop: สร้างโปรเจกต์ TicketPilot และคำสั่ง CLI แสดงรายการ Ticket จำลอง

DAY
2

Ownership, Borrowing และ Collections

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Ownership และการจัดการหน่วยความจำ

- Stack กับ Heap และเหตุผลที่ Rust ไม่ต้องใช้ Garbage Collector
- กฎ Ownership, Move Semantics และการคืนหน่วยความจำด้วย Drop
- Clone กับ Copy และต้นทุนที่แท้จริงของการ clone
- Lab: ทดลอง Move ค่าระหว่างตัวแปรและฟังก์ชันแล้วอ่านผลจาก Compiler

7 Borrowing และ References

- Immutable Reference (&T) และ Mutable Reference (&mut T)
- กฎการยืม และวิธีอ่าน Error ของ Borrow Checker ที่ละบรรทัด
- รูปแบบปัญหาที่พบบ่อยเมื่อเริ่มเขียน Rust และวิธีแก้ด้วยเหตุผล
- Lab: แก้โค้ดที่ Compile ไม่ผ่านโดยไม่ใช้ clone เป็นทางลัด

8 Lifetimes เบื้องต้น

- ความหมายของ Lifetime และกฎ Lifetime Elision
- การระบุ Lifetime ในฟังก์ชันและ Struct ที่เก็บ Reference
- เลือกระหว่างเก็บ Reference กับเก็บข้อมูลแบบเป็นเจ้าของ
- Lab: เขียนฟังก์ชันคืนค่า Reference ที่ Compiler ยอมรับ

9 String, Slice และ Collections

- String กับ &str และการทำงานกับข้อความ UTF-8 รวมถึงภาษาไทย
- Slice ของ Array และ Vector
- Vec, HashMap และ HashSet พร้อมการเลือกใช้ให้เหมาะกับงาน
- วนลูปแบบยืมค่า (iter), เปลี่ยนค่า (iter_mut) และย้ายค่า (into_iter)
- Entry API สำหรับนับและจัดกลุ่มข้อมูล

10 Workshop: ค้นหาและจัดกลุ่ม Ticket

- เขียนฟังก์ชันค้นหาและกรอง Ticket ตามสถานะและความเร่งด่วน
- จัดกลุ่ม Ticket ตามหมวดหมู่ด้วย HashMap และ Entry API
- ออกแบบ Signature ของฟังก์ชันให้รับ Reference แทนการย้ายค่า
- Workshop: ทบทวนโค้ดเพื่อลดการ clone ที่ไม่จำเป็นและเปรียบเทียบผลก่อนหลัง



DAY
3

Traits, Generics, Error Handling และ Testing

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Traits และ Generics

- นิยามและ Implement Trait พร้อม Default Method
- Generics กับ Trait Bounds และ impl Trait
- Trait Object (dyn Trait) และการเลือก Static หรือ Dynamic Dispatch
- Trait มาตรฐานที่ใช้อยู่: Debug, Display, From และ Default
- Lab: แยก Repository ของ Ticket ด้วย Trait

12 Closures และ Iterators

- Closure และการจับค่าตัวแปรจากภายนอก
- Iterator Adapter: map, filter, fold และ collect
- เขียนโค้ดแบบ Functional ให้อ่านง่ายและไม่สร้างข้อมูลซ้ำ
- Lab: เขียนฟังก์ชันจาก Day 2 ใหม่ด้วย Iterator

13 Error Handling อย่างเป็นระบบ

- Result, ตัวดำเนินการ ? และกรณีที่ต้องใช้ panic!
- สร้าง Error Type ของ Library ด้วย thiserror
- ใช้ anyhow ในระดับ Application และเพิ่ม Context ให้ Error
- Lab: สร้าง Error Type ของ TicketPilot และส่งต่อ Error อย่างถูกต้อง

14 Serialization ด้วย serde

- Serialize และ Deserialize ด้วย derive
- อ่านเขียนไฟล์ JSON ด้วย serde_json
- กำหนดชื่อฟิลด์ ค่าเริ่มต้น และฟิลด์ที่ไม่บังคับ
- Lab: โหลดข้อมูล Ticket จากไฟล์ JSON

15 Workshop: Testing ใน Rust

- Unit Test ด้วย #[test] และ cargo test
- Integration Test ในโฟลเดอร์ tests
- ใช้ Trait สร้าง Fake Repository สำหรับทดสอบ
- Workshop: Repository แบบไฟล์ JSON และแบบหน่วยความจำ พร้อม Unit Test ครอบคลุมฟังก์ชันหลัก

DAY
4

Async Rust, Web API และ LLM Integration

เนื้อหาการอบรม · Section 16-20

รวม
6 ชั่วโมง

16 Async Rust กับ Tokio

- async/await, Future และบทบาทของ Runtime
- #[tokio::main], tokio::spawn และการรันงานพร้อมกัน
- แอร์ State ระหว่าง Task ด้วย Arc และ Mutex หรือ RwLock
- Timeout และการยกเลิกงาน
- Lab: เรียก HTTP หลายรายการพร้อมกันด้วย reqwest

17 Lab: สร้าง Web API ด้วย axum

- Router, Handler และ Extractor (Path, Query, Json, State)
- คืนค่า Response และ Error ในรูปแบบ JSON ที่สม่ำเสมอ
- Middleware ด้วย tower เช่น Logging และ Timeout
- Lab: Endpoint ดึงรายการ Ticket และดู Ticket รายตัว

18 พื้นฐานการใช้ LLM สำหรับนักพัฒนา

- Token, Context Window และค่าใช้จ่ายต่อ Request
- ออกแบบ Instructions และ Input สำหรับงานสรุปและจัดหมวดหมู่
- ข้อมูลส่วนบุคคลกับ PDPA และเกณฑ์เลือก Cloud หรือ Local Model
- เหตุผลที่ต้องตรวจสอบผลจาก LLM ก่อนนำไปใช้ทุกครั้ง

19 เชื่อมต่อ OpenAI Responses API และ Ollama

- ติดตั้ง async-openai พร้อม Feature responses และเก็บ API Key อย่างปลอดภัย
- ส่ง Request อ่านผลลัพธ์ และจัดการ Retry กับ Rate Limit
- ติดตั้ง Ollama และโมเดลขนาดเล็กที่รันบน Notebook ได้
- เรียก Ollama ผ่าน HTTP ด้วย reqwest และผ่าน Endpoint ที่เข้ากันได้กับ OpenAI
- Lab: เปรียบเทียบคุณภาพ ความเร็ว และข้อจำกัดของ Cloud กับ Local

20 Workshop: TicketPilot API เรียก LLM

- ออกแบบ Endpoint POST /tickets/{id}/analyze
- ส่ง Ticket ให้ LLM สรุปใจความและจัดหมวดหมู่
- จัดการ Error จาก LLM และ Timeout ให้ API ยังตอบได้
- Workshop: ทดสอบ API ด้วย Ticket จำลองหลายประเภท

DAY
5**GenAI-Powered Rust Service (Master Workshop)**

เนื้อหาการอบรม · Section 21-25

รวม
6 ชั่วโมง**21 Lab: ออกแบบ LLM Provider ด้วย Trait**

- นิยาม Trait LlmProvider และ Implement สำหรับ OpenAI และ Ollama
- เลือก Provider จาก Config ตามข้อกำหนดด้านข้อมูลขององค์กร
- ทดสอบด้วย Fake Provider โดยไม่ต้องเรียก API จริง
- Lab: สลับ Provider โดยไม่แก้ Business Logic

22 Structured Output และการตรวจสอบผลลัพธ์

- กำหนด JSON Schema ของผลลัพธ์: หมวดหมู่ ความเร่งด่วน สรุป และคำตอบแนะนำ
- แปลงผลลัพธ์เป็น Struct ด้วย serde และรับมือกรณีไม่ตรง Schema
- Fallback และ Retry อย่างมีขอบเขต
- Lab: ผลลัพธ์ Ticket ที่ระบบนำไปใช้ต่อได้ทันที

23 Streaming และ Tool Calling เบื้องต้น

- รับผลลัพธ์แบบ Streaming และส่งต่อให้ Client ด้วย Server-Sent Events ใน axum
- แนวคิด Tool Calling: ให้ LLM เรียกฟังก์ชัน Rust เพื่อค้น Ticket ที่เกี่ยวข้อง
- จำกัดสิทธิ์และตรวจ Argument ก่อนรับฟังก์ชันทุกครั้ง
- Lab: Endpoint แบบ Streaming สำหรับร่างคำตอบ

24 ความพร้อมใช้งานจริงและ Docker

- Logging แบบมีโครงสร้างด้วย tracing
- จัดการ Config และ Secret ผ่าน Environment Variables
- Graceful Shutdown และ Health Check Endpoint
- Release Build, การลดขนาด Binary และ Dockerfile แบบ Multi-stage
- Lab: รัน Service คู่กับ Ollama ด้วย Docker Compose

25 Master Workshop: TicketPilot AI Service

- รวม Provider ที่สลับได้, Structured Output และ Endpoint แบบ Streaming
- เพิ่ม Logging, Config และ Unit Test ให้ครบส่วนสำคัญ
- Build เป็น Docker Image และรันร่วมกับ Ollama
- นำเสนอผลงานและสรุปแนวทางนำไปใช้กับระบบขององค์กร

07 เรียนจบแล้วได้อะไร Outcomes

- **พื้นฐาน Rust ที่ถูกต้อง** เข้าใจกลไก ไม่ใช่แค่ Compile ผ่าน
- **เข้าใจ Ownership จริง** อ่านและแก้ Error ของ Borrow Checker ได้
- **โค้ดที่ทดสอบได้** Traits, Generics, Error Type และ cargo test
- **สร้าง Async Web API** Tokio และ axum
- **เชื่อม AI ได้สองแบบ** OpenAI Responses API และ Ollama
- **พร้อม Deploy** Structured Output, tracing และ Docker

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 5 วัน และ Lab Guide รายวันพร้อม Checkpoint สำหรับตามเพื่อนให้ทัน
- Source Code ของ TicketPilot AI Service ทุกขั้นตอน และข้อมูล Ticket จำลองในรูปแบบ JSON
- Template ของ Dockerfile แบบ Multi-stage และ Docker Compose สำหรับรัน Service คู่กับ Ollama
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โขยม ผู้สอนและนักพัฒนาซอฟต์แวร์ที่ใช้ภาษาโปรแกรมสมัยใหม่และ Generative AI กับงานจริง จุดเด่นคือการอธิบายกลไกที่ยากของ Rust อย่าง Ownership และ Borrow Checker ด้วยภาพและตัวอย่างที่เทียบกับภาษาที่คุณเรียนรู้จักอยู่แล้ว แล้วพา ลงมือทำโปรเจกต์ต่อเนื่องจนเห็นผลลัพธ์ด้วยตัวเอง เพื่อให้คุณเรียนกลับไปเขียน Rust และเชื่อมต่อ AI ในงานของตนเองได้อย่างมั่นใจ

ราคาค่าอบรม 11,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

11,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE