สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
PRC-28
PROGRAMMING / RUST

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

Rust Programming

เขียน Rust ให้เข้าใจ Ownership และ สร้าง Web API ด้วย Axum และ PostgreSQL

- 12 ชั่วโมง
- 2 วัน
- พื้นฐาน **เขียนโปรแกรมและเว็บมาบ้าง**
- ระดับ **Beginner-Intermediate**



ระยะเวลา: 2 วัน
12 ชั่วโมง (2 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

Rust เป็นภาษาที่ให้ความเร็วระดับเดียวกับ C และ C++ แต่ป้องกันข้อผิดพลาดด้านหน่วยความจำได้ตั้งแต่ตอนคอมไพล์โดยไม่ต้องมี Garbage Collector จึงถูกใช้มากขึ้นทั้งในงานระบบ เครื่องมือบรรทัดคำสั่ง และ Backend ที่ต้องการทั้งความเร็วและความเสถียร แต่ผู้เริ่มต้นมักติดอยู่กับแนวคิด Ownership, Borrowing และ Lifetime ที่ไม่มีในภาษาอื่น จนไม่ได้ไปถึงการสร้างงานจริง

หลักสูตรนี้ปูพื้นฐาน Rust แบบ Edition 2024 ตั้งแต่การใช้ Cargo ชนิดข้อมูล, Ownership, Borrowing, Struct, Enum, Pattern Matching, การจัดการ Error, Trait, Generic และ Iterator พร้อมเครื่องมือดูแลคุณภาพโค้ดอย่าง Clippy และ rustfmt จากนั้นนำไปสร้าง Web API จริงด้วย Tokio และ Axum เชื่อม PostgreSQL ด้วย SQLx เพิ่มระบบผู้ใช้และการยืนยันตัวตน แล้วปิดท้ายด้วย Workshop สร้าง RESTful Service ที่รันบน Docker (อบรม 2 วัน วันละ 6 ชั่วโมง รวม 12 ชั่วโมง ระดับ Beginner-Intermediate)

เครื่องมือ
- RS Rust Edition 2024
- CG Cargo
- TK Tokio
- AX Axum 0.8
- PG PostgreSQL + SQLx

ราคาค่าอบรม · PUBLIC / IN-HOUSE

7,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide โค้ดตัวอย่างครบทุก Lab ไฟล์ docker-compose สำหรับ PostgreSQL และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

เพิ่มเพื่อน LINE

02-570-8449 · 088-807-9770 · @itgenius · www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- ติดตั้ง Rust ด้วย rustup และจัดการโปรเจกต์ด้วย Cargo ได้คล่อง
- เข้าใจ Ownership, Borrowing และ Lifetime จนอ่านข้อความแจ้งเตือนของคอมไพเลอร์ออก
- ใช้ Struct, Enum, Pattern Matching, Trait และ Generic ออกแบบโค้ดได้
- จัดการ Error ด้วย Result, Option และเครื่องหมาย ? อย่างเป็นระบบ
- เขียนโค้ดแบบ Async ด้วย Tokio และสร้าง Web API ด้วย Axum
- เชื่อมต่อ PostgreSQL ด้วย SQLx พร้อม Migration และ Connection Pool
- เพิ่มระบบผู้ใช้ เข้ารหัสรหัสผ่าน และยืนยันตัวตนด้วย Token
- ทดสอบและแพ็กบริการ Rust เป็น Docker Image ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนาที่ต้องการสร้างเว็บเซอร์วิสด้วยภาษา Rust
- Backend Developer ที่ใช้ Go, Node.js, Java หรือ Python และต้องการภาษาที่เร็วและปลอดภัยกว่าเดิม
- ผู้ที่ต้องการเรียนภาษาที่เน้นประสิทธิภาพและความปลอดภัยของหน่วยความจำ
- วิศวกรระบบที่ต้องการเขียนเครื่องมือหรือบริการที่ทำงานใกล้ระบบมากขึ้น

03 จุดเด่นของหลักสูตร Highlights

- **ปรับปรุงใหม่ปี 2026** Rust Edition 2024 และ Crate รุ่นปัจจุบัน
- **เข้าใจหัวใจของ Rust** Ownership, Borrowing และ Lifetime
- **Web API ยุคใหม่** Tokio, Axum 0.8 และ Tower Middleware
- **ฐานข้อมูลที่ตรวจตอนคอมไพล์** SQLx กับ PostgreSQL และ Migration
- **พร้อมใช้งานจริง** Auth, Test, Clippy และ Docker

04 ความรู้พื้นฐาน Prerequisites

- เขียนโปรแกรมได้อย่างน้อยหนึ่งภาษา เช่น Python, JavaScript, Go, Java หรือ C#
- เข้าใจพื้นฐานเว็บ เช่น HTTP, JSON และ REST API
- ใช้ Command Line และ Git ขั้นพื้นฐานได้
- โน้ตบุ๊กที่ติดตั้ง Rust, VS Code และ Docker Desktop ได้ พร้อมสิทธิ์ติดตั้งโปรแกรม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook Windows, macOS หรือ Linux เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker Desktop ได้
ภาษาและเครื่องมือ	Rust รุ่น Stable ล่าสุดผ่าน rustup, Cargo, Clippy และ rustfmt
Editor	VS Code พร้อม rust-analyzer หรือ RustRover
Web และฐานข้อมูล	Tokio, Axum 0.8, Serde, SQLx และ PostgreSQL บน Docker
ทดสอบ API	ไฟล์ .http ใน VS Code, curl หรือ Postman

หมายเหตุ: ปรับปรุงใหม่ปี 2026 จากฉบับเดิมที่สอน Actix Web กับ Diesel เนื้อหาเปลี่ยนเป็น Rust Edition 2024 ใช้ Axum 0.8 บน Tokio เป็น Web Framework หลัก (สรุปความต่างกับ Actix Web ให้) ใช้ SQLx ที่ตรวจคำสั่ง SQL ตอนคอมไพล์แทน Diesel ปรับหัวข้อ OOP เป็น Trait และ Generic ตามแนวทางของ Rust และเพิ่ม Clippy, การทดสอบ และ Docker เครื่องมือทั้งหมดเป็นโอเพนซอร์สหรือใช้ฟรี RustRover ใช้ฟรีสำหรับงานที่ไม่ใช่เชิงพาณิชย์ หากไม่เข้าเงื่อนไขใช้ VS Code แทนได้ทุก Lab Crate มีการออกรุ่นใหม่บ่อย เนื้อหาอ้างอิงรุ่นล่าสุด ณ วันอบรม

06 ภาพรวมการเรียน 2 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-6	ติดตั้ง Rust และใช้ Cargo, ชนิดข้อมูลและ Control Flow, Ownership และ Borrowing, Struct, Enum และ Pattern Matching, จัดการ Error ด้วย Result และ Trait, Generic, Closure และ Iterator
Day 2 Section 7-12	Module, Crate และการทดสอบ, Async ด้วย Tokio, สร้าง Web API ด้วย Axum, เชื่อม PostgreSQL ด้วย SQLx, ระบบผู้ใช้และการยืนยันตัวตน และ Workshop RESTful Service พร้อม Docker

DAY 1

พื้นฐานภาษา Rust ที่ต้องเข้าใจให้แน่น

เนื้อหาการอบรม · Section 1-6

รวม
6 ชั่วโมง

1 Lab: เริ่มต้น Rust และ Cargo

- Rust เหมาะกับงานแบบใด และจุดต่างจาก C++, Go และภาษาที่มี Garbage Collector
- ติดตั้งด้วย rustup, รอบออกรุ่นทุก 6 สัปดาห์ และ Edition 2024
- cargo new, build, run, check และ cargo build --release
- Lab: โปรแกรมแรกพร้อมจัดรูปแบบด้วย rustfmt และตรวจด้วย Clippy

2 ชนิดข้อมูลและ Control Flow

- ตัวแปร, mut, Shadowing และ Type Inference
- String กับ &str และการทำงานกับข้อความ UTF-8 รวมถึงภาษาไทย
- if, loop, while, for และ Let Chain ใน Edition 2024
- Lab: โปรแกรมคำนวณค่าขนส่งจากข้อมูลที่ผู้ใช้ป้อน

3 Ownership, Borrowing และ Lifetime

- Move, Clone และ Copy ต่างกันอย่างไร
- Reference แบบ &T และ &mut T และกฎของ Borrow Checker
- Slice และ Lifetime เบื้องต้นในฟังก์ชัน
- Lab: แก้โค้ดที่คอมไพล์ไม่ผ่านจากข้อความแจ้งเตือนของคอมไพเลอร์

4 Lab: Struct, Enum และ Pattern Matching

- Struct, Method และ Associated Function
- Enum ที่เก็บข้อมูลได้ และ Option แทนค่า null
- match, if let และ let-else
- Lab: โมเดลข้อมูลคำสั่งซื้อพร้อมสถานะด้วย Enum

5 จัดการ Error อย่างเป็นระบบ

- panic กับ Result และเมื่อไรควรใช้แต่ละแบบ
- เครื่องหมาย ? สำหรับส่ง Error ต่อ
- สร้างชนิด Error ด้วย thiserror และใช้ anyhow ในระดับแอปพลิเคชัน
- Lab: อ่านไฟล์ CSV และรายงาน Error ที่อ่านเข้าใจได้

6 Lab: Trait, Generic, Closure และ Iterator

- Trait แทนการสืบทอด Class และ Trait มาตรฐานอย่าง Debug, Display และ From
- Generic และ Trait Bound
- Closure และ Iterator: map, filter, fold และ collect
- Lab: สรุปยอดขายจากรายการคำสั่งซื้อด้วย Iterator

DAY 2

สร้าง Web API ด้วย Rust

เนื้อหาการอบรม · Section 7-12

รวม
6 ชั่วโมง

7 Module, Crate และการทดสอบ

- จัดโค้ดด้วย mod, pub และ use
- Library Crate กับ Binary Crate และ Cargo Workspace
- Unit Test, Integration Test และ cargo test
- Lab: แยก Business Logic เป็น Library พร้อม Test

8 Async ด้วย Tokio

- แนวคิด async และ await, Runtime, #[tokio::main] และ tokio::spawn
- แชร์ข้อมูลระหว่าง Task ด้วย Arc และ Mutex อย่างปลอดภัย
- เรียก API ภายนอกแบบ Async ด้วย reqwest
- Lab: ดึงข้อมูลจากหลาย API พร้อมกันและรวมผลลัพธ์

9 Lab: Web API ด้วย Axum

- Router, Handler และ Path Parameter แบบ {id} ของ Axum 0.8
- Extractor: Path, Query, Json และ State พร้อมแปลง JSON ด้วย Serde
- Middleware จาก tower-http: Logging, CORS และ Timeout
- Lab: API จัดการสินค้าแบบเก็บข้อมูลในหน่วยความจำ

10 Lab: PostgreSQL ด้วย SQLx

- รัน PostgreSQL บน Docker, สร้าง PgPool และ Migration ด้วย sqlx-cli
- query! และ query_as! ที่ตรวจคำสั่ง SQL ตอนคอมไพล์ และ Offline Mode
- เพิ่ม ดึง แก้ไข และลบข้อมูล พร้อม Transaction
- Lab: ย้าย API สินค้าไปเก็บข้อมูลใน PostgreSQL

11 ระบบผู้ใช้และการยืนยันตัวตน

- ออกแบบ User Model และเก็บรหัสผ่านด้วย Argon2
- Login และออก Token แบบ JWT
- ตรวจ Token ด้วย Extractor หรือ Middleware ก่อนเข้าถึง Route
- Lab: ป้องกัน API เพิ่ม แก้ไข และลบสินค้าให้เฉพาะผู้ใช้ที่ Login

12 Workshop: RESTful Service พร้อม Docker

- Workshop: ออกแบบ Endpoint ของบริการสั่งซื้อสินค้าจากโจทย์
- เขียน Integration Test ให้ API ด้วยฐานข้อมูลทดสอบ
- แพ็กเป็น Docker Image ขนาดเล็กแบบ Multi-stage Build
- สรุปแนวทางต่อยอด เช่น Actix Web, gRPC และ WebAssembly

07 เรียนจบแล้วได้อะไร Outcomes

- **ใช้ Rust และ Cargo คล่อง** rustup, Clippy และ rustfmt
- **เข้าใจ Ownership** Borrowing และ Lifetime ไม่ติดคอมไพเลอร์
- **ออกแบบโค้ดแบบ Rust** Struct, Enum, Trait และ Generic
- **จัดการ Error เป็นระบบ** Result, ?, thiserror และ anyhow
- **สร้าง Web API ได้** Tokio, Axum และ SQLx กับ PostgreSQL
- **พร้อมใช้งานจริง** JWT, Test และ Docker Image

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 2 วัน และ Lab Guide ทีละขั้นตอน
- โค้ดตัวอย่างครบทุก Lab และโปรเจกต์ RESTful Service จาก Workshop
- ไฟล์ docker-compose สำหรับ PostgreSQL และ Dockerfile แบบ Multi-stage
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สนิทวงศ์ กมลภากรณ์ ผู้สอนและนักพัฒนาระบบที่ใช้ Rust และภาษาฝั่ง Backend หลายภาษาในงานจริง จุดเด่นคือการอธิบาย Ownership และ Borrowing ผ่านข้อความจากคอมไพเลอร์ทีละขั้น ให้ผู้เรียนเข้าใจว่าทำไม Rust จึงบังคับกฎเหล่านี้ แล้วพาไปสร้าง Web API ที่ใช้งานได้จริงในเวลาสั้น เพื่อให้กลับไปเริ่มโปรเจกต์ Rust ของตนเองได้อย่างมั่นใจ

ราคาค่าอบรม 7,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

7,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449 · 088-807-9770

@itgenius · contact@itgenius.co.th

เพิ่มเพื่อน LINE