



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
AIC-66

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

AI / AGENT ENGINEERING

AI Harness Engineering

ออกแบบและสร้าง Runtime ของ AI Agent ด้วย Coding Agent

● 12 ชั่วโมง

● 2 วัน

● ต้องเขียน Python ได้

● ระดับ **Advanced**



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

เมื่อ AI Agent ต้องทำงานยาวหลายสิบขั้น เรียกเครื่องมือจริง แก่ไฟล์จริง และรับคำสั่งแบบระบบจริง สิ่งที่ต้องสนใจว่า Agent จะนำเช็ทหรือไม่ได้ อยู่ที่ตัวโมเดลอย่างเดียว แต่อยู่ที่ Harness หรือ Runtime ที่ห่อโมเดลไว้ ทั้ง Loop ที่รู้ว่าเมื่อไรต้องหยุด เครื่องมือที่บอก Error ชัดพอให้ Agent แก้ตัวเองได้ Policy ที่บังคับใช้ด้วยโค้ดแทนการขอร้องใน Prompt และการจัดการ Context ที่ไม่ทำให้ Agent เสื่อมเมื่อทำงานนานขึ้น

หลักสูตรนี้พาวีศวกที่ใช้ AI ช่วยเขียนโค้ดอยู่แล้วก้าวจากผู้ใช้ไปเป็นผู้สร้างระบบ ผู้เรียนจะผ่าโครงสร้างของ Coding Agent เชิงพาณิชย์เพื่อเข้าใจว่า Harness ต่างจาก Framework อย่างไร แล้วเขียน Spec ของแต่ละส่วนให้ละเอียดพอที่ Coding Agent อย่าง Claude Code หรือ Cursor จะสร้างให้ได้ ตั้งแต่ Agent Loop ที่มีเพดาน Step และ Token, Tool Layer, Hooks และ Middleware ไปจนถึง Context Engineering, Subagents, Sandbox และ Checkpoint สำหรับงานที่รันยาว ระหว่างทางฝึกอ่านโค้ดที่ AI เขียนให้เห็นจุดบกพร่องเชิงออกแบบที่ Test จับไม่ได้ และปิดท้ายด้วย Capstone บน Repository จริงที่ผู้เรียนต้องให้เหตุผลกับการตัดสินใจเชิงสถาปัตยกรรมที่ขัดกับคำแนะนำของ Agent (อบรม 2 วัน วันละ 6 ชั่วโมง รวม 12 ชั่วโมง ระดับ Advanced)

เครื่องมือ



Claude Code



Python



Agent SDK



Docker



OpenTelemetry

ราคาค่าอบรม · PUBLIC / IN-HOUSE

4,900 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide ไฟล์ Spec ตัวอย่าง Source Code ของ Harness คนทุก Workshop และไฟล์ docker-compose พร้อมใช้ อาหารกลางวันและอาหารว่าง และ Certificate · ผู้เรียนเตรียมบัญชี Coding Agent และ LLM API Key มาเอง · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- เขียน Spec ของส่วนต่าง ๆ ใน Agent Runtime ให้ละเอียดพอที่ Coding Agent นำไปสร้างได้ถูกต้อง
- มองเห็นจุดบกพร่องเชิงออกแบบในโค้ดที่ AI เขียน ซึ่ง Test ทัวไปตรวจไม่พบ
- สร้างและควบคุม Agent Loop ด้วยเพดาน Step และ Token พร้อมเงื่อนไขการหยุด
- ออกแบบ Tool Interface และ Error Surface ที่ช่วยให้ Agent แก้ข้อผิดพลาดได้เอง
- ใช้ Hooks และ Middleware บังคับ Policy แบบ Deterministic แทนการพึ่ง Prompt
- จัดการ Context ด้วย Compaction, Tool-result Clearing และ External Memory โดยวัด Trade-off ได้
- ใช้ Sandbox และ Checkpoint ให้ Agent ที่ทำงานยาวกู้คืนจากความล้มเหลวได้
- ให้เหตุผลกับการตัดสินใจเชิงสถาปัตยกรรม และประเมินหรือปรับใช้ส่วนประกอบของ Agent SDK ได้

02 กลุ่มเป้าหมาย Target Audience

- AI Engineer ที่อยากก้าวจากผู้ใช้ AI ไปเป็นผู้สร้างระบบ Agent
- Backend Engineer ที่ต้องต่อ Agent เข้ากับระบบและเครื่องมือจริง
- Tech Lead ที่ต้องเลือกสถาปัตยกรรมของ Agent ให้ทีม
- Architect ที่ต้องตรวจทานงานที่ AI เขียนในระดับการออกแบบ
- Platform Engineer ที่ต้องดูแลและบำรุงรักษา Agent ที่รันในองค์กร

03 จุดเด่นของหลักสูตร Highlights

- สร้าง Harness เอง Loop, Tools, Hooks และ Context ครบชุด
- เขียน Spec ให้ Agent สร้าง ฝึกสั่ง Coding Agent ด้วย Spec ที่แม่นยำ
- Policy ด้วยโค้ด Hooks และ Middleware แทนการขอใน Prompt
- รองรับงานรันยาว Compaction, Sandbox และ Checkpoint
- ไม่ผูกกับเจ้าเดียว ใช้ LLM หลายค่ายสลับกันได้

04 ความรู้พื้นฐาน Prerequisites

- เคยใช้ AI Chat หรือ Coding Agent ช่วยเขียนโค้ดมาแล้ว หลักสูตรนี้ไม่เหมาะกับผู้ที่มองหาเครื่องมือแบบ No-code
- เขียน Python ได้คล่อง และเข้าใจ async / await ของ Python 3.12
- ใช้ Git, Command Line และ Docker ขั้นพื้นฐานได้
- เตรียมบัญชี Coding Agent (Claude Code หรือ Cursor) และ LLM API Key ที่มีเครดิต

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker และ Docker Compose ได้
Coding Agent	Claude Code และ Cursor สำหรับสร้างส่วนต่าง ๆ ของ Harness จาก Spec
ภาษาและ SDK	Python 3.12 แบบ async และ Agent SDK
LLM	LLM หลายผู้ให้บริการ เพื่อไม่ผูกระบบไว้กับเจ้าใดเจ้าหนึ่ง
Runtime และ Observability	Docker, Docker Compose, SQLite สำหรับ Checkpoint และ OpenTelemetry

หมายเหตุ: ผู้เรียนต้องเตรียมบัญชี Coding Agent และ LLM API Key ที่มีเครดิตมาเอง ค่าใช้ API เป็นไปตามการใช้งานจริง เครื่องมือด้าน Runtime ทั้งหมดเป็นโอเพนซอร์สและรันบนเครื่องของผู้เรียนด้วย Docker หลักสูตรนี้เน้นการสร้าง Runtime ของ Agent ไม่ได้สอนพื้นฐานการใช้ Coding Agent ผู้ที่ต้องการปูพื้นฐานการใช้ Agent ในงานพัฒนาซอฟต์แวร์ก่อน แนะนำหลักสูตร AI Coding Agent for SDLC

06 ภาพรวมการเรียนรู้ 2 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-6	โครงสร้างของ Harness, Agent Loop ที่มีเพดาน Step และ Token, Tool Layer ที่ให้ Agent แก่ตัวเองได้ และ Hooks กับ Middleware ที่บังคับ Policy แบบ Deterministic พร้อม Workshop ให้ Coding Agent ลงมือสร้างจาก Spec
Day 2 Section 7-12	Context Engineering ด้วย Compaction, Tool-result Clearing และ External Memory, Subagents, Sandbox กับ Checkpoint สำหรับงานยาว, ขอบเขตของ Agent SDK และ Capstone uu Repository จริง



DAY
1

Runtime Core: Loop, Tools และ Hooks

เนื้อหาการอบรม · Section 1-6

รวม
6 ชั่วโมง

1 ภายวิภาคของ Harness

- Harness ต่างจาก Framework อย่างไร และใครควบคุม Control Flow
- องค์ประกอบของระบบ Agentic AI: Model, Loop, Tools, Context และ Policy
- ฝ่าโครงสร้าง Coding Agent เชิงพาณิชย์ว่าแต่ละส่วนทำหน้าที่อะไร
- ตั้งค่า Claude Code และ Cursor ที่ใช้ในห้องเรียนให้พร้อมทำงานจาก Spec

2 Agent Loop

- วงจร Observe, Decide, Act และ Verify ของ Agent
- กำหนดงบ Step และงบ Token ต่อหนึ่งงาน
- เชื้อไขการหยุด และการตรวจจบ Loop ที่วนซ้ำไม่จบ
- Structured Tracing ให้ย้อนดูได้ว่า Agent ตัดสินใจอะไรในแต่ละรอบ

3 Workshop: สร้าง Agent Loop จาก Spec

- เขียน Spec ของ Loop ที่ระบุเพดาน Step และ Token ชัดเจน
- กำหนดเชื้อไขการหยุดและรูปแบบ Trace ที่ต้องการ
- ให้ Coding Agent สร้าง Loop ตาม Spec ด้วย Python async
- ตรวจโค้ดที่ได้ หากดูบกพร่องเชิงออกแบบที่ Test ไม่เจอ แล้วสั่งแก้

4 Tool Layer

- คำอธิบาย Tool คือส่วนหนึ่งของ Prompt ที่ต้องออกแบบอย่างตั้งใจ
- Schema ที่ชัดเจนจนโมเดลไม่ต้องเดาค่าพารามิเตอร์
- Error Surface ที่บอกสาเหตุและทางแก้ ให้ Agent แก้ตัวเองได้
- Idempotency สำหรับ Tool ที่มีผลข้างเคียง และการตัดทอนผลลัพธ์ที่ใหญ่เกินก่อนส่งกลับเข้า Context

5 Workshop: Tool อ่านไฟล์ แก้ไฟล์ และรับคำสั่ง

- เขียน Spec ของ Tool อ่านไฟล์ แก้ไฟล์ และรับคำสั่ง
- ออกแบบ Error Message ที่ Agent นำไปแก้ไขต่อได้
- ให้ Coding Agent สร้าง Tool และต่อเข้ากับ Loop จาก Workshop แรก
- ทดสอบกรณีผิดพลาดและผลลัพธ์ขนาดใหญ่

6 Hooks และ Middleware

- Policy ต้องอยู่ใน Runtime ไม่ใช่ฝากไว้ใน Prompt
- Lifecycle Hooks ที่ทำงานแบบ Deterministic ก่อนและหลังการเรียก Tool
- ลำดับของ Middleware Chain และผลเมื่อสลับลำดับ
- Workshop: ให้ Coding Agent สร้าง Middleware Chain จาก Spec โดยไม่แตะโค้ดของ Loop



DAY
2

Context Engineering และการกู้คืนระบบ

เนื้อหาการอบรม · Section 7-12

รวม
6 ชั่วโมง

7 Context Engineering

- ทำโม Agent ที่ทำงานยาวจึงเชื่อมประสิทธิภาพลงเรื่อย ๆ
- Tool-result Clearing: ล้างผลลัพธ์เก่าที่ไม่ต้องใช้แล้ว
- Compaction: ย่อประวัติให้เป็นบล็อกที่มีโครงสร้าง
- External Memory ที่อยู่รอบแม่ Context ถูกใช้
- ทำไมต้องใช้ทั้งสามวิธีร่วมกัน และวัด Trade-off อย่างไร

8 Workshop: ลงมือทำ Context Engineering

- เขียน Spec ของ Tool-result Clearing, Compaction และ External Memory
- ให้ Coding Agent สร้างทั้งสามส่วนและต่อเข้ากับ Harness
- วัดผลด้วย Trace ก่อนและหลัง ทั้งจำนวน Token และคุณภาพงาน

9 Subagents และการแยก Context

- แยก Context ของ Subagent แต่ละตัวออกจากกัน
- สัญญาการส่งผลกลับแบบสั้น พร้อม Pointer ไปยังรายละเอียด
- ความเสี่ยงเมื่อแต่ละส่วนลด Context เองโดยไม่ประสานกัน
- เลือกว่าเมื่อไรควรแตก Subagent และเมื่อไรไม่ควร

10 Sandbox และการกู้คืน

- จำกัดสิทธิ์ Filesystem, Network และ Process ด้วย Docker
- Retry แบบไม่ทำซ้ำสิ่งที่สำเร็จไปแล้ว
- Checkpoint และ Resume ด้วย SQLite สำหรับงานที่รันยาว
- Cost Cap และ Kill Switch เมื่อ Agent ใช้ทรัพยากรเกินกำหนด
- ส่ง Trace ด้วย OpenTelemetry เพื่อติดตามงานที่ล้มเหลว

11 ขอบเขตของ Agent SDK และการนำไปใช้ในโดเมนอื่น

- เทียบ Agent SDK กับ Harness ที่สร้างเอง
- เทรนด์ที่ตัดสินว่าจะต่อยอดของเดิมหรือสร้างใหม่
- ประเมินและปรับใช้ส่วนประกอบของ Agent SDK
- นำความรู้ไปใช้กับ Agent งานบริการลูกค้า งานข้อมูล และ RAG

12 Capstone: Coding Agent บน Repository จริง

- ใช้ Harness และ Coding Agent ทำงานบน Repository จริง
- ตรวจสอบที่ Agent เขียนด้วยมุมมองเชิงสถาปัตยกรรม
- นำเสนอการตัดสินใจ 3 ข้อที่ขัดกับคำแนะนำของ Agent พร้อมเหตุผล
- สรุปแนวทางนำ Harness ไปใช้ในงานของทีม

07 เรียนจบแล้วได้อะไร Outcomes

- เข้าใจ Harness ทั้งระบบ Loop, Tools, Hooks และ Context
- เขียน Spec ให้ Agent สร้าง ละเอียดพอให้ Coding Agent ทำได้ถูก
- ตรวจสอบโค้ด AI ได้ลึก เห็นจุดบกพร่องที่ Test จับไม่ได้
- บังคับ Policy ด้วยโค้ด Hooks และ Middleware แบบ Deterministic
- Agent ทำงานยาวได้ Context Engineering, Sandbox, Checkpoint
- ตัดสินใจเชิงสถาปัตยกรรม เลือก SDK หรือสร้างเองอย่างมีเหตุผล

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 2 วัน และ Lab Guide ที่ละเอียด
- ไฟล์ Spec ตัวอย่างของทุกส่วนใน Harness ที่นำไปสร้าง Coding Agent ต่อได้
- Source Code ของ Harness คนทุก Workshop และไฟล์ docker-compose พร้อมใช้
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โทยม วิศวกรและผู้สอนที่ออกแบบและสร้างระบบ AI Agent สำหรับใช้งานจริง ทั้งงานพัฒนาซอฟต์แวร์และงานเชื่อมระบบขององค์กร จุดเด่นคือการสอนให้ผู้เรียนคิดแบบผู้ออกแบบระบบ ไม่ใช่แค่ผู้สั่ง AI โดยให้ผู้เรียน Spec เอง อ่านโค้ดที่ Agent สร้างอย่างมีวิจารณญาณ และกล้าตัดสินใจต่างจากคำแนะนำของ AI เมื่อมีเหตุผลเชิงสถาปัตยกรรมรองรับ

ราคาค่าอบรม 4,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

4,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

☎ 02-570-8449

📞 @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE