สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
WDC-87
PROGRAMMING / C#

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

C# OOP and Design Patterns

ออกแบบโค้ด C# ให้ดูแลง่ายด้วยหลัก OOP, SOLID และ Design Patterns

- 18 ชั่วโมง
- 3 วัน
- พื้นฐาน **เขียนภาษา C# พื้นฐานได้**
- ระดับ **Intermediate**



ระยะเวลา: 3 วัน
18 ชั่วโมง (3 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

เมื่อเขียน C# ได้แล้ว ปัญหาที่ตามมาเมื่อระบบใหญ่ขึ้นคือโค้ดแก้ยาก เพิ่มฟีเจอร์ทีไรกระทบส่วนอื่น และนำกลับมาใช้ซ้ำไม่ได้ การคิดแบบ Object Oriented และการรู้จัก Design Pattern ที่นักพัฒนาทั่วโลกใช้แก้ปัญหาซ้ำ ๆ จะช่วยให้ออกแบบระบบที่แยกส่วนชัดเจน ทดสอบได้ และปรับเปลี่ยนได้ง่ายขึ้น

หลักสูตรนี้เป็นฉบับปรับปรุงใหม่สำหรับ C# 14 บน .NET 10 (LTS) ต่อยอดจากหลักสูตร Basic C# Programming เริ่มจากหลักการ OOP ทั้ง 4 ข้อ Class และ Object แบบ C# สมัยใหม่ Interface, Generics และ Event ไปจนถึงหลัก SOLID และ Dependency Injection แล้วลงมือเขียน Design Patterns ที่ใช้บ่อยทั้งกลุ่ม Creational, Structural และ Behavioral พร้อมเขียน Unit Test ด้วย xUnit และ Workshop ออกแบบระบบจากโจทย์ธุรกิจ (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

- C# 14
- .NET 10 LTS
- Visual Studio 2026
- Dependency Injection
- xUnit

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide ซอร์สโค้ดตัวอย่างครบทุก Pattern และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

เพิ่มเพื่อน LINE

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

01 วัตถุประสงค์ Objectives

- เข้าใจหลักการคิดแบบ Object Oriented และเหตุผลที่ใช้แทนการเขียนแบบ Top-Down
- ออกแบบ Class ด้วย Encapsulation, Abstraction, Inheritance และ Polymorphism ได้
- ใช้ฟีเจอร์ C# สมัยใหม่ เช่น Primary Constructor, Record, คีย์เวิร์ด field และ Extension Members ได้
- เลือกใช้ abstract class, sealed และ Interface ได้อย่างเหมาะสม
- นำหลัก SOLID และ Dependency Injection ไปใช้ลดการผูกติดกันของโค้ดได้
- เขียน Design Patterns ที่ใช้บ่อยทั้งกลุ่ม Creational, Structural และ Behavioral ได้
- ทดสอบโค้ดเชิงวัตถุด้วย Unit Test และ xUnit ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนาที่เขียน C# ได้แล้วและต้องการเขียนโค้ดที่ดูแลง่ายขึ้น
- ผู้ที่ต้องการพัฒนาเว็บแอปพลิเคชันหรือ API ด้วย C# และ ASP.NET Core
- IT Manager และหัวหน้าทีมที่ต้องการวางมาตรฐานการออกแบบโค้ดของทีม
- นักศึกษาที่ผ่านพื้นฐาน C# มาแล้วและต้องการเข้าใจ OOP อย่างลึกซึ้ง

03 จุดเด่นของหลักสูตร Highlights

- **อัปเดตเป็น C# 14** .NET 10 LTS และ Visual Studio 2026
- **OOP ครบ 4 หลักการ** พร้อมฟีเจอร์ภาษาสมัยใหม่
- **SOLID และ DI** แนวทางเดียวกับ ASP.NET Core
- **Design Patterns ที่ใช้จริง** Creational, Structural และ Behavioral
- **เขียน Test ไปด้วย** ยืนยันการออกแบบด้วย xUnit

04 ความรู้พื้นฐาน Prerequisites

- เขียนโปรแกรมด้วยภาษา C# ได้ หรือผ่านหลักสูตร Basic C# Programming มาแล้ว
- เข้าใจตัวแปร เงื่อนไข การวนซ้ำ Method และ Collection ใน C#
- เคยใช้ Visual Studio หรือ VS Code มาบ้าง
- โน้ตบุ๊กที่มีสิทธิ์ติดตั้งโปรแกรม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook Windows, macOS หรือ Linux เชื่อมต่ออินเทอร์เน็ต พร้อมสิทธิ์ติดตั้งโปรแกรม
แพลตฟอร์ม	.NET 10 SDK (LTS) และภาษา C# 14
เครื่องมือพัฒนา	Visual Studio 2026 Community บน Windows หรือ VS Code พร้อม C# Dev Kit ทุกระบบปฏิบัติการ
Library และการทดสอบ	Microsoft.Extensions.DependencyInjection, xUnit, dotnet CLI และ Git
ออกแบบ	UML Class Diagram ด้วย Mermaid หรือเครื่องมือวาดไดอะแกรมที่ผู้เรียนถนัด

หมายเหตุ: ปรับปรุงใหม่ปี 2026 จากหลักสูตร C# OOP เดิม: ใช้ C# 14 บน .NET 10 (LTS) เพิ่มฟีเจอร์ภาษาที่เปลี่ยนวิธีเขียน OOP เช่น Primary Constructor, Record, required, คีย์เวิร์ด field, Default Interface Method และ Extension Members เพิ่มหลัก SOLID, Dependency Injection และ Unit Test ปรับ Singleton ให้ใช้ Lazy และ Lifetime ของ DI แทนการเขียนแบบเดิมที่ไม่ปลอดภัยเมื่อทำงานหลาย Thread และสอน Observer ด้วย event ของ C# ต่อยอดจากหลักสูตร Basic C# Programming ซึ่งสอน Class และ Inheritance ระดับพื้นฐาน เครื่องมือทั้งหมดใช้ได้ฟรี Visual Studio Community มีเงื่อนไขสิทธิ์การใช้งานสำหรับองค์กร หากไม่เข้าเงื่อนไขใช้ VS Code แทนได้ทุก Lab เนื้อหาอิงเวอร์ชันล่าสุด ณ วันอบรม

06 ภาพรวมการเรียน 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	แนวคิด OOP และเตรียม .NET 10, Class และ Object แบบ C# 14, Encapsulation ด้วย Property และคีย์เวิร์ด field, Abstraction และการออกแบบ Class ด้วย UML และ Inheritance เทียบกับ Composition
Day 2 Section 6-10	Polymorphism และ Overloading, sealed และ abstract class, Interface สมัยใหม่, Generics, Delegate, Event และ Extension Members และหลัก SOLID กับ Dependency Injection
Day 3 Section 11-15	Design Patterns ของ Gang of Four กลุ่ม Creational, Structural และ Behavioral ได้แก่ Singleton, Factory Method, Composite, Decorator, Observer, Strategy, Command และ State แล้วปิดท้ายด้วย Workshop พร้อม Unit Test

DAY 1

หลักคิด OOP และการออกแบบ Class

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 Lab: ทำไมต้อง OOP และเตรียมเครื่องมือ

- ปัญหาของโค้ดแบบ Top-Down เมื่อระบบใหญ่ขึ้น และข้อดีข้อเสียของ OOP
- หลักการคิดแบบ Object: ข้อมูลและพฤติกรรมอยู่ด้วยกัน
- ติดตั้ง .NET 10 SDK และ Visual Studio 2026 หรือ VS Code กับ C# Dev Kit
- สร้าง Solution ที่มี Class Library, Console App และ Test Project ด้วย dotnet CLI
- Lab: แปลงโปรแกรมแบบ Top-Down ให้เป็น Class แรก

2 Lab: Class และ Object ใน C# 14

- Field, Property, Method และ Constructor
- Primary Constructor และ Object Initializer
- สมาชิกแบบ required และ init สำหรับข้อมูลที่ต้องกำหนดตอนสร้าง
- สมาชิกแบบ static และเมื่อไรควรใช้
- Lab: สร้าง Class สินค้าและตะกร้าสินค้า

3 Lab: Encapsulation

- ห่อหุ้มสถานะของ Object ด้วย Access Modifier
- Property ต่างจาก Field อย่างไร และการตรวจค่าก่อนกำหนด
- คีย์เวิร์ด field ใน C# 14 สำหรับ Property ที่มี Logic
- Immutability ด้วย Record และ readonly
- Lab: ป้องกันไม่ให้ยอดเงินในบัญชีถูกแก้ไขผิดวิธี

4 Lab: Abstraction และการออกแบบ Class

- หลักการนามธรรม: แสดงเฉพาะสิ่งที่ผู้ใช้ Object ต้องรู้
- แบ่งหน้าที่ของ Class ให้ชัดเจนจากโจทย์ธุรกิจ
- อ่านและวาด UML Class Diagram ด้วย Mermaid
- Value Object กับ Entity ต่างกันอย่างไร
- Lab: ออกแบบ Class ของระบบจองห้องประชุม

5 Lab: Inheritance และ Composition

- หลักการสืบทอดด้วย base class และคีย์เวิร์ด base
- Access Modifier แบบ protected และลำดับการเรียก Constructor
- ความสัมพันธ์แบบ is-a กับ has-a
- เมื่อไรควรเลือก Composition แทน Inheritance
- Lab: ปรับโครงสร้าง Class พนักงานหลายประเภท

DAY 2

Polymorphism, Interface และ SOLID

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: Polymorphism

- หลักการ Polymorphism หรือการมีได้หลายรูปแบบ
- Method Overloading และ Operator Overloading
- virtual, override และความต่างกับการซ่อนด้วย new
- ตรวจชนิดของ Object ด้วย Pattern Matching
- Lab: คำนวณค่าขนส่งหลายรูปแบบผ่าน Method เดียวกัน

7 Lab: sealed และ abstract class

- ความสำคัญของ sealed class และ sealed method
- abstract class และ abstract method ต่างจาก class ทั่วไปอย่างไร
- Template ของขั้นตอนด้วย abstract class
- Lab: ระบบชำระเงินที่บังคับให้ทุกช่องทางทำตามขั้นตอนเดียวกัน

8 Lab: Interface

- การใช้งาน Interface และความต่างระหว่าง abstract class กับ Interface
- Implement หลาย Interface และ Explicit Implementation
- Default Interface Method และ Static Abstract Member
- Interface มาตรฐานของ .NET: IComparable, IEquatable และ IDisposable
- Lab: เรียงลำดับและเปรียบเทียบ Object ด้วย Interface

9 Lab: Generics, Delegate, Event และ Extension Members

- Generic Class และ Method พร้อม Constraint
- Delegate, Func, Action และ Lambda
- ประกาศและใช้งาน event สำหรับแจ้งการเปลี่ยนแปลง
- เพิ่มความสามารถให้ Type เดิมด้วย Extension Members แบบ C# 14
- Lab: Repository แบบ Generic ที่แจ้ง Event เมื่อข้อมูลเปลี่ยน

10 Lab: หลัก SOLID และ Dependency Injection

- Single Responsibility และ Open/Closed Principle
- Liskov Substitution, Interface Segregation และ Dependency Inversion
- Dependency Injection ด้วย Microsoft.Extensions.DependencyInjection
- Lifetime แบบ Singleton, Scoped และ Transient
- Lab: ปรับโค้ดที่ผูกติดกันให้ทดสอบได้ด้วย DI

DAY 3

Design Patterns และ Workshop

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: Design Patterns และกลุ่ม Creational

- Design Pattern คืออะไร และองค์ความรู้ของ Gang of Four
- Singleton แบบปลอดภัยด้วย Lazy เทียบกับ Singleton Lifetime ของ DI
- Factory Method สำหรับสร้าง Object ตามเงื่อนไข
- Builder สำหรับ Object ที่มีขั้นตอนการสร้างหลายขั้น
- Lab: สร้างตัวส่งการแจ้งเตือนหลายช่องทางด้วย Factory Method

12 Lab: Structural Patterns

- Composite สำหรับข้อมูลแบบลำดับชั้น เช่น เมนูและโครงสร้างองค์กร
- Adapter สำหรับเชื่อมโค้ดใหม่กับระบบเดิม
- Decorator สำหรับเพิ่มความสามารถโดยไม่แก้ Class เดิม
- Facade สำหรับซ่อนความซับซ้อนของหลายระบบย่อย
- Lab: เพิ่ม Logging และ Cache ให้ Service ด้วย Decorator

13 Lab: Observer และ Strategy

- Observer Pattern ด้วย event และ IObservable
- Strategy Pattern สำหรับสลับอัลกอริทึมขณะทำงาน
- ใช้ Strategy ร่วมกับ Dependency Injection
- Lab: ระบบคำนวณส่วนลดหลายเงื่อนไขด้วย Strategy

14 Lab: Behavioral Patterns อื่น ๆ

- Command Pattern สำหรับคำสั่งที่ย้อนกลับ (Undo) ได้
- Template Method สำหรับขั้นตอนที่มีโครงเดียวกัน
- State Pattern สำหรับ Object ที่เปลี่ยนพฤติกรรมตามสถานะ
- เลือก Pattern ให้เหมาะ และระวังการใช้ Pattern เกินจำเป็น
- Lab: สถานะคำสั่งซื้อด้วย State Pattern

15 Workshop: ออกแบบระบบจากโจทย์ธุรกิจพร้อม Unit Test

- วิเคราะห์โจทย์และวาด Class Diagram ของระบบ
- เลือกใช้หลัก OOP, SOLID และ Pattern ที่เหมาะกับแต่ละส่วน
- เขียน Unit Test ด้วย xUnit เพื่อยืนยันพฤติกรรมของระบบ
- นำเสนองาน รีวิวการออกแบบร่วมกัน และแนวทางต่อยอดสู่ ASP.NET Core

07 เรียนจบแล้วได้อะไร Outcomes

- **คิดแบบ OOP ได้** ออกแบบ Class จากโจทย์จริง
- **ใช้ 4 หลักการ OOP ได้** Encapsulation ถึง Polymorphism
- **ใช้ C# สมัยใหม่ได้** Record, field และ Extension Members
- **เขียนโค้ดที่ยืดหยุ่น** Interface, SOLID และ DI
- **ใช้ Design Patterns ได้** Factory, Decorator, Observer และ Strategy
- **ทดสอบการออกแบบได้** Unit Test ด้วย xUnit

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- ซอร์สโค้ดตัวอย่างครบทุก Lab และ Design Pattern พร้อม Unit Test
- สรุปการเลือกใช้ Design Pattern และ Checklist หลัก SOLID สำหรับรีวิวโค้ด
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและนักพัฒนาซอฟต์แวร์ที่ใช้ C# และ .NET ในงานจริง อธิบายหลัก OOP และ Design Pattern ผ่านโจทย์ที่เห็นภาพ เริ่มจากโค้ดที่มีปัญหาแล้วค่อย ๆ ปรับให้ดีขึ้น พร้อมบอกเหตุผลและข้อแลกเปลี่ยนของแต่ละแนวทาง เพื่อให้ผู้เรียนเลือกใช้ได้อย่างเหมาะสมแทนการท่องจำ และนำไปใช้กับโปรเจกต์ของทีมได้ทันที

ราคาค่าอบรม 9,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

9,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449 · 088-807-9770

@itgenius · contact@itgenius.co.th

เพิ่มเพื่อน LINE