

(Computer Programming for Computer Engineer)

รหัสรายวิชาเดิม : ไม่มี

วิชาบังคับก่อน : ENGCE306 การเขียนโปรแกรมคอมพิวเตอร์

ศึกษาและฝึกปฏิบัติการเกี่ยวกับหลักการเขียนโปรแกรมเชิงลึกเกี่ยวกับ พอยน์เตอร์และอาร์เรย์ การจองหน่วยความจำ ไฟล์อินพุต-เอาต์พุต ฟังก์ชันเรียกตัวเอง การเขียนและออกแบบโปรแกรมเชิงวัตถุ การซ่อนข้อมูล การสืบทอด การพ้องรูป คลาสนามธรรม การเขียนโปรแกรมแบบหลายเทรต การพัฒนาส่วนติดต่อกับผู้ใช้ การทดสอบซอฟต์แวร์ เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์ ระบบควบคุมเวอร์ชัน

Study and practice of advanced programming concepts on pointers and arrays, memory allocation, file I/O, recursive function, object-oriented programming and design, encapsulation, inheritance, polymorphism, abstract class, multithreaded programming, graphical user interface, software testing, software development tools, version control systems.

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมสำหรับวิศวกรคอมพิวเตอร์

(Computer Programming for Computer Engineer)

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
1	ทบทวนพื้นฐาน Python และแนะนำ OOP	- แบบทดสอบก่อนเรียน (Pre-test) - กิจกรรม: วิเคราะห์โค้ด Python พื้นฐานเป็นกลุ่ม
2	ตัวแปรอ้างอิง (Reference), List และ Array	- ปฏิบัติ: ใช้งาน list, array, และดู memory address ด้วย id() - การบ้าน: เขียนโปรแกรมค้นหาและเรียงลำดับข้อมูล
3	การจัดการหน่วยความจำ และโครงสร้างข้อมูลลิงก์	- ปฏิบัติ: สร้างคลาส Node และ Singly Linked List - Quiz ครั้งที่ 1: list, reference, หน่วยความจำ
4	การอ่านและเขียนไฟล์ (File I/O)	- ปฏิบัติ: อ่านไฟล์ .txt, เขียน log ไฟล์ - การบ้าน: วิเคราะห์ข้อมูลจากไฟล์ CSV และสรุปผล
5	ฟังก์ชันเรียกตัวเอง (Recursion)	- ปฏิบัติ: เขียน Fibonacci และ Factorial แบบ recursive - Quiz ครั้งที่ 2: การทำงานของ recursion และ stack

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมสำหรับวิศวกรคอมพิวเตอร์

(Computer Programming for Computer Engineer) (ต่อ)

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
6	การสร้างคลาสและการซ่อนข้อมูล (Encapsulation)	- ปฏิบัติ: สร้าง class จำลองระบบบัญชีธนาคาร - การบ้าน: เขียนคลาสที่มี constructor, method และ attribute
7	การสืบทอดและการพ้องรูป (Inheritance & Polymorphism)	- ปฏิบัติ: สร้างคลาสหลักและคลาสย่อย เช่น Vehicle → Car, Boat - การบ้าน: เขียนระบบจัดการสัตว์ในสวนสัตว์
8	คลาสนามธรรม (Abstract Class) และ Interface	- ปฏิบัติ: ใช้ abc module และ @abstractmethod - แฉกงาน Mini Project: สร้างระบบจัดการข้อมูลโดยใช้ OOP + File I/O
9	การเขียนโปรแกรมแบบหลายเทร็ด (Multithreading)	- ปฏิบัติ: โปรแกรมจับเวลา + โปรแกรมอื่นรันพร้อมกัน - Quiz ครั้งที่ 3: แนวคิด Thread / GIL / Parallel
10	การสร้าง GUI ด้วย Tkinter	- ปฏิบัติ: เขียน GUI โปรแกรมคำนวณ หรือฟอร์มลงทะเบียน - การบ้าน: สร้าง GUI รับข้อมูลผู้ใช้และแสดงผล
11	การทดสอบซอฟต์แวร์ (Software Testing)	- ปฏิบัติ: สร้าง Test case ด้วย unittest หรือ pytest - Quiz ครั้งที่ 4: หลักการเขียน test, assert, coverage
12	เครื่องมือพัฒนาซอฟต์แวร์ + ระบบควบคุมเวอร์ชัน (Git)	- ปฏิบัติ: ใช้งาน Git และ GitHub - การบ้าน: ส่งโปรเจกต์ผ่าน Pull Request
13	เวิร์กช็อปโปรเจกต์ปลายภาค (Final Project Workshop)	- นักศึกษาลงมือพัฒนา Final Project - อาจารย์ตรวจความคืบหน้า + ให้คำแนะนำรายกลุ่ม
14	การนำเสนอ Final Project + สรุปทเรียน	- นักศึกษา Demo ผลงาน + ตอบคำถาม - แบบทดสอบหลังเรียน (Post-test) + ประเมินตนเอง

ตารางที่ 2 การประเมินผลรายวิชา การเขียนโปรแกรมสำหรับวิศวกรคอมพิวเตอร์

(Computer Programming for Computer Engineer)

ประเภทการประเมินผล	รายละเอียด	สัดส่วน (%)
งานเดี่ยว / การบ้าน (Assignment)	<ul style="list-style-type: none"> - งานเดี่ยวรายสัปดาห์ เช่น โปรแกรมค้นหาข้อมูล, วิเคราะห์ไฟล์, เขียนคลาส, GUI ฟอรัม - ตรวจสอบความถูกต้องของโค้ด, ความครบถ้วน, ความเข้าใจในแนวคิด และการส่งงานผ่าน GitHub หรือระบบ LMS 	25%
แบบทดสอบย่อย / กิจกรรมในห้อง (Quiz & Lab)	<ul style="list-style-type: none"> - Quiz 3-4 ครั้ง เช่น พอยน์เตอร์ (reference), recursion, thread, test case - กิจกรรมในห้อง เช่น Debugging, เขียน Test, ทำงานกลุ่มเขียนคลาส/โค้ดร่วมกัน 	15%
มินิโปรเจกต์ (สอบกลางภาค)	<ul style="list-style-type: none"> - พัฒนาแอปขนาดเล็ก เช่น ระบบจัดการข้อมูล (CLI หรือ GUI) - ต้องใช้แนวคิด OOP + File I/O และอาจมีการแสดงผลผ่าน GUI (Tkinter) 	20%
โปรเจกต์ปลายภาค (Final Project + การนำเสนอ)	<ul style="list-style-type: none"> - พัฒนาแอปพลิเคชันเต็มรูปแบบแบบเดี่ยวหรือกลุ่ม (2-4 คน) - ใช้แนวคิดครอบคลุมทั้ง OOP, GUI, Thread, Test, และ Version Control - นำเสนอหน้าชั้น พร้อมตอบคำถาม และมี GitHub repo 	30%
จิตพิสัย / วินัยการเรียน	<ul style="list-style-type: none"> - การมาเรียนตรงเวลาและสม่ำเสมอ - การส่งงานตรงเวลา - การมีส่วนร่วมในห้องเรียน/กลุ่ม - ความมีวินัย ความรับผิดชอบ 	10%
รวม		100%