

## (Computer Programming)

รหัสรายวิชาเดิม : ไม่มี

วิชาบังคับก่อน : ไม่มี

ศึกษาและฝึกปฏิบัติเกี่ยวกับองค์ประกอบคอมพิวเตอร์, การปฏิสัมพันธ์ระหว่างฮาร์ดแวร์กับซอฟต์แวร์, ชนิดข้อมูล, อัลกอริทึม, รหัสเทียม และผังงาน, ตัวดำเนินการ, คำสั่งควบคุมแบบทางเลือก, คำสั่งควบคุมแบบวนรอบ, ตัวแปรชุด, ฟังก์ชัน, การประยุกต์ใช้เพื่อแก้ปัญหาในงานทางวิศวกรรม, แนวโน้มภาษาคอมพิวเตอร์สมัยใหม่

Study and practice of computer components, hardware and software interaction, data types, algorithm, pseudo code and flowchart, operators, selection statements, iteration statements, arrays, functions, problem solving in engineering, modern programming language trends.

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมคอมพิวเตอร์

ลำดับ	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
1	แนะนำรายวิชาและโครงสร้างพื้นฐานของระบบคอมพิวเตอร์	<ul style="list-style-type: none"> <li>แนะนำภาพรวมของรายวิชาและข้อตกลงในชั้นเรียน</li> <li>ติดตั้งเครื่องมือพัฒนา เช่น VS Code, Thonny, Google Colab</li> <li>ทำแบบทดสอบก่อนเรียนเพื่อสำรวจพื้นฐาน</li> </ul>
2	การใช้งานภาษา Python เบื้องต้นและชนิดข้อมูลพื้นฐาน	<ul style="list-style-type: none"> <li>ฝึกเขียนโปรแกรมโดยใช้คำสั่ง input(), print()</li> <li>ใช้งานชนิดข้อมูล int, float, str, และ bool</li> </ul>
3	ตัวแปร นิพจน์ และตัวดำเนินการ	<ul style="list-style-type: none"> <li>เขียนโปรแกรมคำนวณพื้นฐานทางคณิตศาสตร์</li> <li>ฝึกใช้ตัวดำเนินการทางตรรกะและเปรียบเทียบ</li> </ul>
4	การเขียนรหัสเทียม (Pseudocode) และผังงาน (Flowchart)	<ul style="list-style-type: none"> <li>เขียนผังงานโดยใช้ Draw.io หรือ Lucidchart</li> <li>แปลงรหัสเทียมให้เป็นภาษา Python</li> </ul>
5	คำสั่งควบคุมแบบทางเลือก (If / If-Else)	<ul style="list-style-type: none"> <li>เขียนโปรแกรมเพื่อจำลองเงื่อนไข เช่น การคำนวณเกรด</li> <li>แบบทดสอบย่อย: โครงสร้างควบคุมแบบทางเลือก</li> </ul>
6	คำสั่งควบคุมแบบวนรอบ (Looping)	<ul style="list-style-type: none"> <li>เขียนโปรแกรมด้วยคำสั่ง for และ while เพื่อแก้โจทย์ซ้ำ</li> <li>การประยุกต์ใช้คำสั่งซ้อน</li> </ul>
7	การประยุกต์ใช้โครงสร้างควบคุมร่วมกัน	<ul style="list-style-type: none"> <li>ฝึกเขียนโปรแกรมที่มีหลายเงื่อนไข เช่น โปรแกรม ATM, โปรแกรมเงินทอน</li> <li>การบ้าน: ออกแบบโปรแกรมจากสถานการณ์จริง</li> </ul>

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมคอมพิวเตอร์ (ต่อ)

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
8	การแก้ปัญหาโดยใช้แนวคิดเชิงอัลกอริทึมการแก้ปัญหาโดยใช้แนวคิดเชิงอัลกอริทึม	<ul style="list-style-type: none"> <li>กำหนดปัญหา ออกแบบแนวทางแก้ไข และเขียนเป็นโค้ด Python</li> </ul>
9	สอบกลางภาค	<ul style="list-style-type: none"> <li>ดำเนินการสอบกลางภาค (ข้อเขียนหรือภาคปฏิบัติตามความเหมาะสม)</li> </ul>
10	โครงสร้างข้อมูลแบบลำดับ (List / Tuple)	<ul style="list-style-type: none"> <li>ฝึกใช้งาน list และ tuple ในการจัดเก็บข้อมูล</li> <li>เขียนโปรแกรมจัดการข้อมูลสินค้า คะแนน ฯลฯ</li> </ul>
11	การใช้ List 2 มิติ และการประยุกต์ใช้งาน	<ul style="list-style-type: none"> <li>พัฒนาโปรแกรมประมวลผลข้อมูลตาราง เช่น Matrix</li> <li>การบ้าน: โปรแกรมคำนวณเงินเดือนแบบตาราง</li> </ul>
12	ฟังก์ชันและการแยกส่วนของโปรแกรม	<ul style="list-style-type: none"> <li>เขียนฟังก์ชันด้วยคำสั่ง def และการส่งค่าพารามิเตอร์</li> <li>แบบทดสอบย่อย: โครงสร้างของฟังก์ชัน</li> </ul>
13	ตัวแปรระดับต่าง ๆ และการส่งค่าระหว่างฟังก์ชัน	<ul style="list-style-type: none"> <li>อธิบาย Global และ Local Variables</li> <li>การบ้าน: ฟังก์ชันหาค่าเฉลี่ย ค่าสูงสุด ฯลฯ</li> </ul>
14	การประยุกต์ใช้ภาษา Python ในงานวิศวกรรมเบื้องต้น	<ul style="list-style-type: none"> <li>เขียนโปรแกรมทางด้านไฟฟ้า กลศาสตร์ หรือการคำนวณทั่วไป เช่น ความเร็ว แรงดัน</li> </ul>
15	การอ่าน/เขียนไฟล์ และแนวคิดการจัดการข้อมูลเบื้องต้น	<ul style="list-style-type: none"> <li>ใช้คำสั่ง open() และ with open() เพื่ออ่าน/บันทึกข้อมูล</li> <li>บันทึกผลการคำนวณลงในไฟล์ข้อความ</li> </ul>
16	โครงงานปลายภาค (Final Project)	<ul style="list-style-type: none"> <li>วางแผนและพัฒนาโครงงาน เช่น ระบบจัดเกรด เครื่องคิดเลข GUI</li> </ul>
17	การนำเสนอและสาธิตโครงงานปลายภาค	<ul style="list-style-type: none"> <li>กลุ่มนักเรียนนำเสนอและสาธิตระบบที่พัฒนา พร้อมตอบคำถาม</li> </ul>
18	สอบปลายภาค	<ul style="list-style-type: none"> <li>ดำเนินการสอบปลายภาค (ภาคทฤษฎีหรือปฏิบัติตามลักษณะรายวิชา)</li> </ul>

ตารางที่ 2 การประเมินผลรายวิชา การเขียนโปรแกรมคอมพิวเตอร์

องค์ประกอบการประเมินผล	รายละเอียด	สัดส่วน (%)
งานเดี่ยวและแบบฝึกหัดรายสัปดาห์	<ul style="list-style-type: none"> <li>• แบบฝึกหัดการเขียนโปรแกรม ฟังก์ชัน และโจทย์เชิงปฏิบัติ</li> <li>• พิจารณาจากความครบถ้วน ความถูกต้อง ความคิดสร้างสรรค์ และการส่งงานตรงเวลา</li> </ul>	15%
แบบทดสอบย่อยและกิจกรรมในชั้นเรียน	<ul style="list-style-type: none"> <li>• แบบทดสอบสั้น (2-3 ครั้ง) ในหัวข้อสำคัญ เช่น Control Flow, Function, File</li> <li>• กิจกรรมกลุ่ม: วิเคราะห์โค้ดและแก้ไขปัญหา</li> </ul>	15%
สอบกลางภาค	<ul style="list-style-type: none"> <li>• การสอบวัดผลความเข้าใจเนื้อหาครั้งแรกของรายวิชา</li> </ul>	20%
สอบปลายภาค	<ul style="list-style-type: none"> <li>• การสอบวัดผลความเข้าใจเนื้อหาครั้งสุดท้ายของรายวิชา</li> </ul>	20%
โครงการปลายภาค (Final Project)	<ul style="list-style-type: none"> <li>• พัฒนาโปรแกรมในลักษณะกลุ่ม โดยบูรณาการเนื้อหาที่เรียน</li> <li>• ประเมินจากคุณสมบัติของโปรแกรม ความคิดสร้างสรรค์ เอกสาร และการนำเสนอ</li> </ul>	20%
คุณลักษณะนิสัยและวินัยในการเรียนรู้	<ul style="list-style-type: none"> <li>• การเข้าชั้นเรียนตรงเวลาและสม่ำเสมอ</li> <li>• ความรับผิดชอบในการส่งงาน</li> <li>• การมีส่วนร่วมในชั้นเรียนและการทำงานกลุ่ม</li> <li>• ความมีวินัยและความรับผิดชอบต่อตนเองและผู้อื่น</li> </ul>	10%
รวม		100%