การเขียนโปรแกรมคอมพิวเตอร์สำหรับคณิตศาสตร์ 1

Computer Programming for Mathematics 1 | รหัสวิชา 040223101

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคการศึกษาที่ 1/2569

ข้อมูลรายวิชา

รหัสและชื่อวิชา

040223101 การเขียนโปรแกรมคอมพิวเตอร์สำหรับคณิตศาสตร์ 1 (Computer Programming for Mathematics 1)

หน่วยกิต

3(2-2-5) – บรรยาย 2 ชั่วโมง / ปฏิบัติ 2 ชั่วโมง / ศึกษาด้วยตนเอง 5 ชั่วโมงต่อสัปดาห์

วิชาบังคับก่อน

ไม่มี

คำอธิบายรายวิชา (ภาษาไทย)

กระบวนการแก้ปัญหา การจำลองแนวคิด ทักษะการแก้ปัญหา การเขียนโปรแกรมเบื้องต้น กระบวนการของโปรแกรม ชนิดข้อมูล ตัวดำเนินการเปรียบเทียบ ตัวดำเนินการเชิงตรรกะ หลักการเขียนโปรแกรมและเทคนิคในการแก้ปัญหาทางคณิตศาสตร์

คำอธิบายรายวิชา (English)

Problem solving process; simulation of ideas; problem solving skills; basic programming; program procedure; data type; comparison operator; logical operators; programming principles and techniques for solving mathematical problems.

จุดมุ่งหมายของรายวิชา

1. **หลักการคอมพิวเตอร์และ Python**
   อธิบายหลักการทำงานของคอมพิวเตอร์ การประมวลผลข้อมูล และกระบวนการพัฒนาโปรแกรมด้วยภาษา Python ได้

2. **การเขียนโปรแกรมพื้นฐาน**
   เขียนโปรแกรมโดยใช้ตัวแปร ชนิดข้อมูล การรับ/แสดงผล และไลบรารีสำเร็จรูปได้อย่างถูกต้อง

3. **โครงสร้างการตัดสินใจและการทำซ้ำ**
   ออกแบบและเขียนโปรแกรมที่ใช้โครงสร้างการตัดสินใจและการทำซ้ำเพื่อแก้ปัญหาได้

4. **โครงสร้างข้อมูลและฟังก์ชัน**
   ประยุกต์โครงสร้างข้อมูล (ลิสต์ ดิกชันนารี เซต) และฟังก์ชันในการจัดการข้อมูลและสร้างโปรแกรมแบบมอดูลได้

5. **การคำนวณเชิงตัวเลขด้วย NumPy**
   ใช้การคำนวณเชิงตัวเลขด้วย NumPy เพื่อแก้ปัญหาทางคณิตศาสตร์ได้

ตำราและเอกสารประกอบการสอน

ตำราอ่านประกอบหลัก

หนังสือสอนเขียนโปรแกรมภาษา Python ของคณะวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

เอกสารประกอบการสอน

มีเอกสารประกอบการสอนรายวิชา 040223101 การเขียนโปรแกรมคอมพิวเตอร์สำหรับคณิตศาสตร์ 1 โดยเฉพาะ

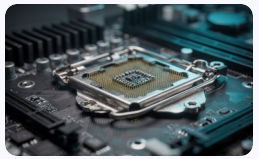
นักศึกษาสามารถใช้ตำราหรือหนังสืออื่น ๆ ทั้งภาษาไทยและภาษาอังกฤษ ซึ่งมีเนื้อหาตรงตามที่กำหนดได้เพิ่มเติม

แหล่งเรียนรู้เพิ่มเติม

นักศึกษาได้รับการสนับสนุนให้ค้นคว้าจากแหล่งข้อมูลที่หลากหลาย ทั้งภาษาไทยและภาษาอังกฤษ เพื่อเสริมสร้างความเข้าใจในเนื้อหาที่กำหนดในรายวิชา

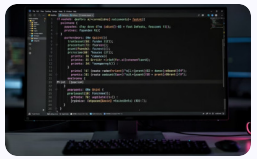
สัปดาห์ที่ 1 - 2

Introduction to Computers, Python, Variables & I/O



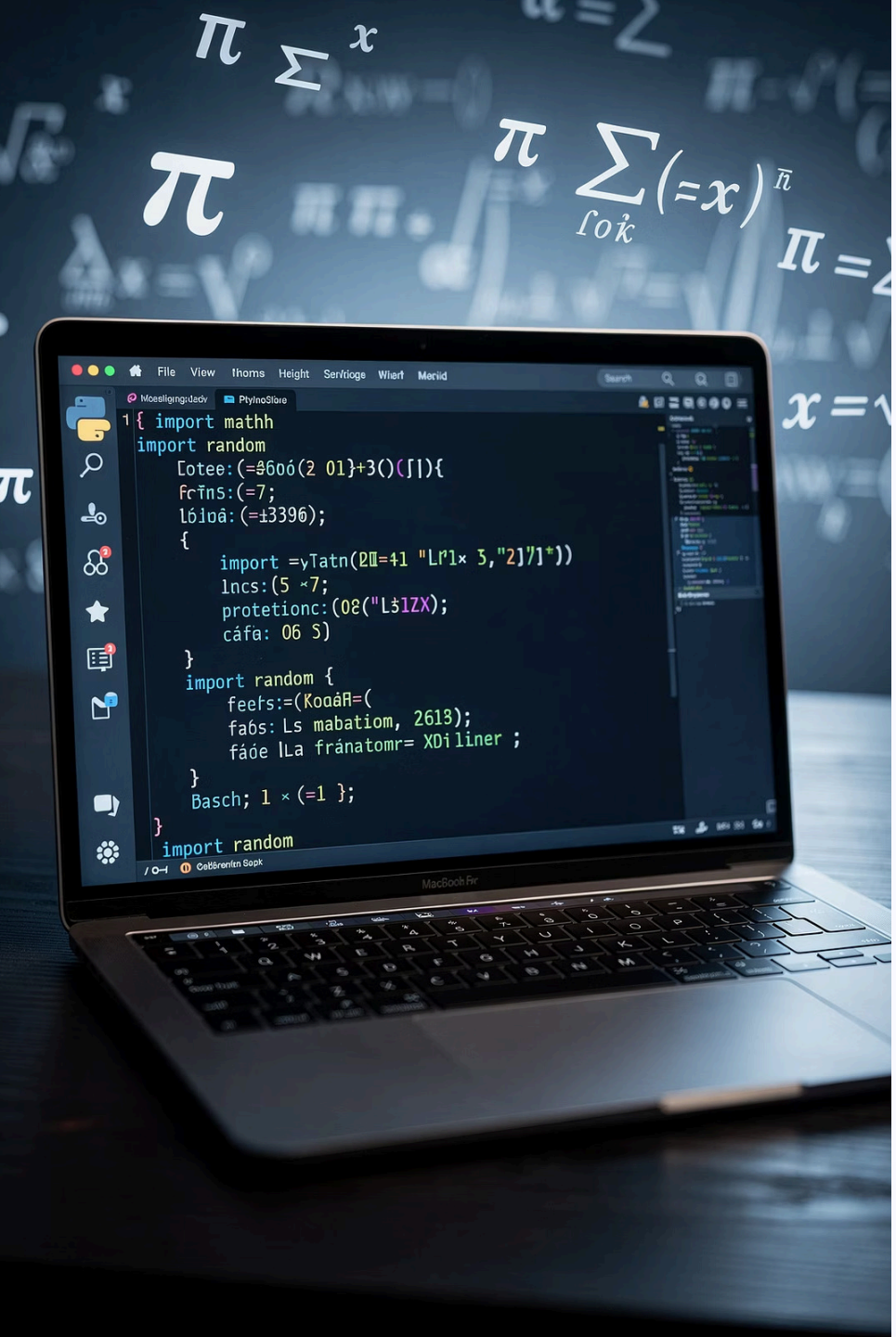
สัปดาห์ที่ 1 – Introduction to Computers and Python

ศึกษาองค์ประกอบและสถาปัตยกรรมคอมพิวเตอร์ แนวคิดการประมวลผลข้อมูล การติดตั้งและใช้งาน Python รวมถึงการเขียนและรันโปรแกรมเบื้องต้นเป็นครั้งแรก



สัปดาห์ที่ 2 – Variables, Input and Output

เรียนรู้เรื่องตัวแปร ชนิดข้อมูลพื้นฐาน การกำหนดค่า การใช้คำสั่ง `input()` และ `print()` รวมถึงการจัดรูปแบบผลลัพธ์ให้แสดงผลได้อย่างถูกต้องและสวยงาม



สัปดาห์ที่ 3

Libraries – ไลบรารีสำเร็จรูปใน Python

นักศึกษาจะได้เรียนรู้การนำไลบรารีสำเร็จรูปมาใช้งานในโปรแกรม Python โดยเริ่มจากการ import โมดูลต่าง ๆ

การ import โมดูล

วิธีการนำเข้าโมดูลและไลบรารีเข้าสู่โปรแกรม Python ด้วยคำสั่ง import และ from...import

ไลบรารี math

ฟังก์ชันทางคณิตศาสตร์ เช่น math.sqrt(), math.pi, math.log() และอื่น ๆ

ไลบรารี random

การสุ่มตัวเลขและข้อมูลด้วย random.random(), random.randint() สำหรับการจำลองทางคณิตศาสตร์

สัปดาห์ที่ 4 - 5

Conditional Structure – โครงสร้างการตัดสินใจ

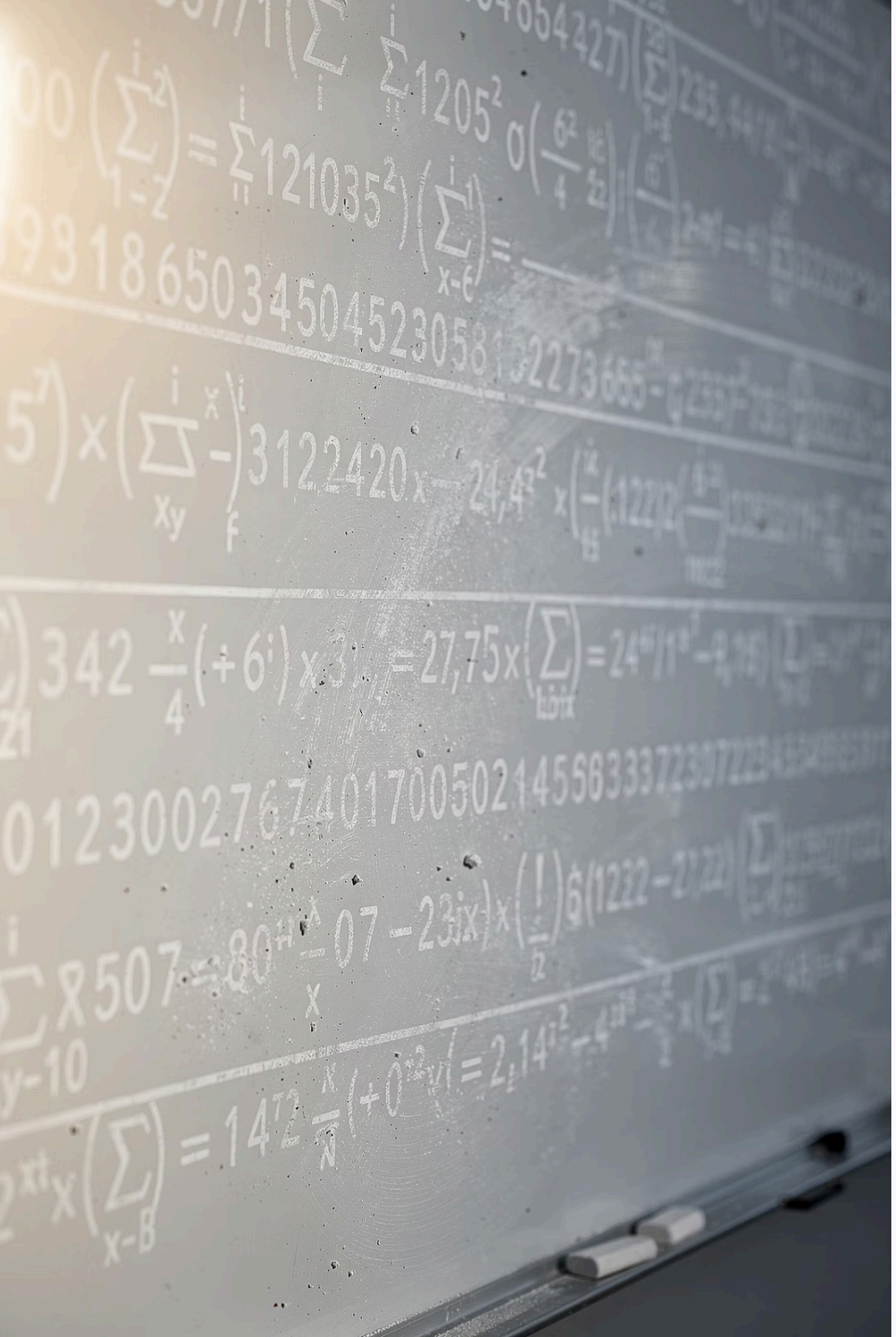
สัปดาห์ที่ 4 – พื้นฐาน Conditional

เรียนรู้การใช้คำสั่ง if / elif / else นิพจน์เงื่อนไขและตรรกะ การเยื้อง (indentation) และบล็อกคำสั่งที่ถูกต้องตามหลักการของ Python

สัปดาห์ที่ 5 – เงื่อนไขซ้อนและโจทย์ประยุกต์

ฝึกปฏิบัติการเขียนเงื่อนไขซ้อน (nested conditions) และนำไปประยุกต์กับโจทย์ทางคณิตศาสตร์จริง เช่น การตรวจสอบจำนวนเฉพาะ การแก้สมการ และการจำแนกประเภทข้อมูล

- เงื่อนไขซ้อน (Nested Conditions)
- ตัวดำเนินการเปรียบเทียบและตรรกะ
- โจทย์ประยุกต์ทางคณิตศาสตร์



สัปดาห์ที่ 6

Repetition Structure: For Loops

โครงสร้างการทำซ้ำด้วย `for` loop เป็นเครื่องมือสำคัญในการแก้ปัญหาทางคณิตศาสตร์ที่ต้องการการคำนวณซ้ำ ๆ

คำสั่ง for และ range()

การใช้ `for` loop ร่วมกับ `range()` เพื่อกำหนดจำนวนรอบการวนซ้ำ

การวนซ้ำและการคำนวณ

การคำนวณผลรวม ผลคูณ และอนุกรมทางคณิตศาสตร์ด้วยการวนซ้ำ

การประยุกต์ใช้งาน

นำ for loop ไปใช้แก้โจทย์คณิตศาสตร์ เช่น การหาผลรวมอนุกรม การสร้างตาราง และการตรวจสอบเงื่อนไขซ้ำ

สัปดาห์ที่ 7

Lists – โครงสร้างข้อมูลแบบลิสต์

การสร้างและเข้าถึงลิสต์

การสร้างลิสต์ การเข้าถึงสมาชิกด้วย index การ slicing เพื่อดึงข้อมูลบางส่วน และการแก้ไขค่าในลิสต์

เมท็อดของลิสต์

การใช้เมท็อดสำคัญ เช่น `append()`, `remove()`, `sort()`, `len()` และอื่น ๆ เพื่อจัดการข้อมูลในลิสต์

การวนซ้ำกับลิสต์

การใช้ `for` loop ร่วมกับลิสต์เพื่อประมวลผลข้อมูลทีละสมาชิก การประยุกต์ใช้ลิสต์ในการเก็บชุดข้อมูลทางคณิตศาสตร์ เช่น ชุดตัวเลข เวกเตอร์ และลำดับ

ลิสต์เป็นโครงสร้างข้อมูลพื้นฐานที่สำคัญที่สุดใน Python และเป็นรากฐานของการเรียนรู้ NumPy ในภายหลัง

สัปดาห์ที่ 8

While Loops – การวนซ้ำแบบมีเงื่อนไข

คำสั่ง while

การใช้ `while` loop เพื่อวนซ้ำตราบเท่าที่เงื่อนไขยังเป็นจริง เหมาะสำหรับกรณีที่ไม่ทราบจำนวนรอบล่วงหน้า

เงื่อนไขหยุดการวน

การกำหนดเงื่อนไขหยุดที่ชัดเจน การใช้ `break` และ `continue` เพื่อควบคุมการไหลของโปรแกรม

การหลีกเลี่ยงการวนไม่รู้จบ

เทคนิคการป้องกัน infinite loop การตรวจสอบเงื่อนไขและการอัปเดตตัวแปรควบคุมอย่างถูกต้อง

การประยุกต์เชิงตัวเลข

นำ while loop ไปใช้ในการแก้ปัญหาเชิงตัวเลข เช่น การหาค่าลู่เข้า การทำซ้ำจนกว่าจะได้ความแม่นยำที่ต้องการ

สอบกลางภาค

ครอบคลุมเนื้อหาสัปดาห์ที่ 1 - 8 ทั้งหมด

Introduction to Python

องค์ประกอบคอมพิวเตอร์, การติดตั้ง, การเขียนและรันโปรแกรม

Variables & I/O

ตัวแปร, ชนิดข้อมูล, `input()`, `print()`

Libraries

การ import โมดูล, ไลบรารี math และ random

Conditional & Loops

if/elif/else, for loops, while loops, Lists

สัปดาห์ที่ 9

Dictionaries and Sets – ดิกชันนารีและเซต

Dictionaries – ดิกชันนารี

โครงสร้างข้อมูลแบบคู่ key - value ที่ช่วยให้เข้าถึงข้อมูลได้อย่างรวดเร็ว การสร้าง การเข้าถึง การแก้ไข และการลบข้อมูล รวมถึงเมท็อดสำคัญ เช่น `.keys()`, `.values()`, `.items()`

Sets – เซต

โครงสร้างข้อมูลที่เก็บสมาชิกไม่ซ้ำกัน การดำเนินการเซตทางคณิตศาสตร์ ได้แก่ ยูเนียน (`|`), อินเตอร์เซกชัน (`&`), ผลต่าง (`-`) และการตรวจสอบความสัมพันธ์ระหว่างเซต

สัปดาห์ที่ 10 - 11

Functions – ฟังก์ชันและการเขียนโปรแกรมแบบมอดูล

01

การนิยามฟังก์ชัน (สัปดาห์ที่ 10)

การใช้คำสั่ง `def` เพื่อนิยามฟังก์ชัน การกำหนดพารามิเตอร์และอาร์กิวเมนต์ การใช้ `return` เพื่อคืนค่า และการทำความเข้าใจขอบเขตตัวแปร (scope) ทั้งแบบ local และ global

02

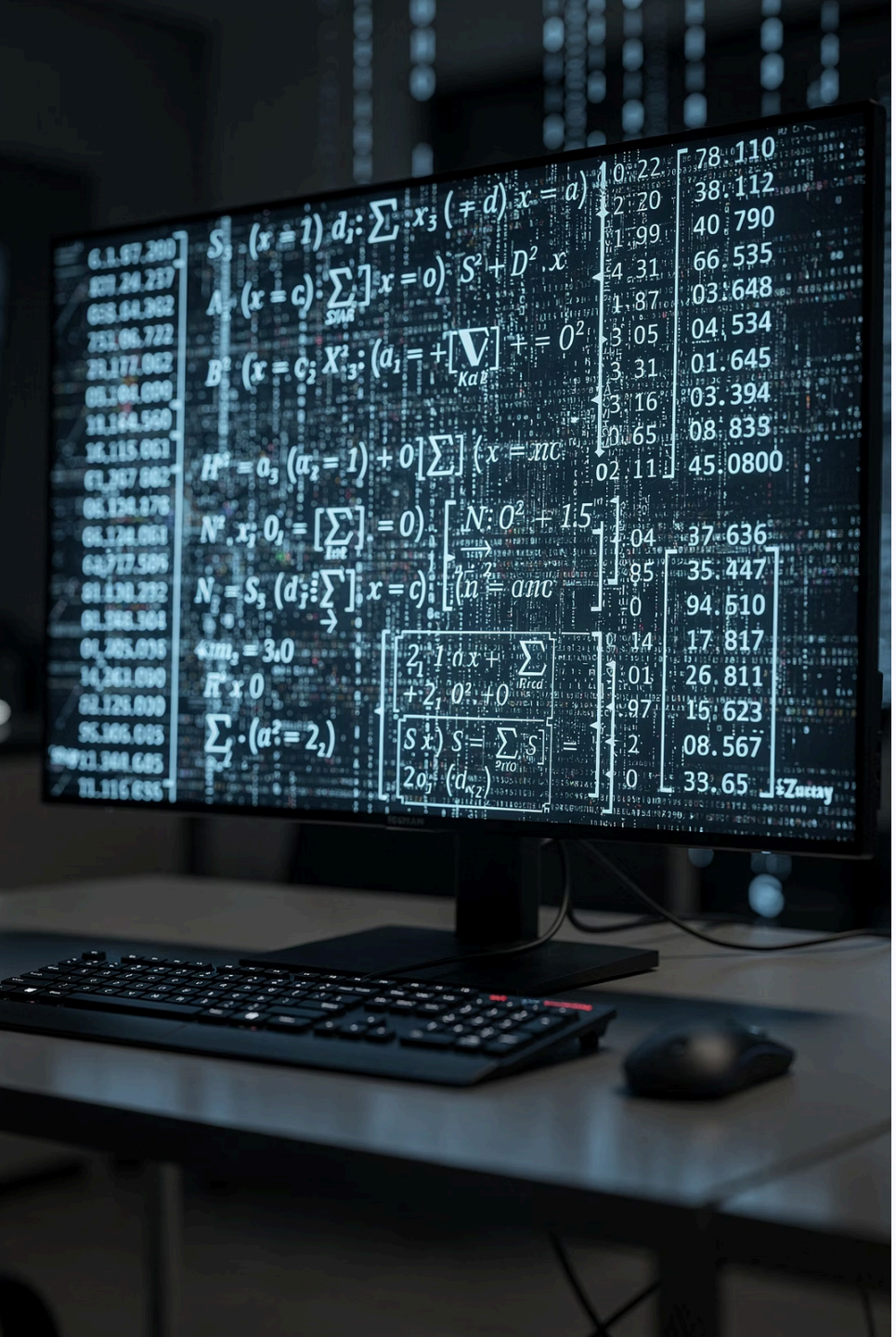
ฟังก์ชันขั้นสูง (สัปดาห์ที่ 11)

ค่าปริยาย (default parameters) การคืนค่าหลายค่าพร้อมกัน การเขียนฟังก์ชันทางคณิตศาสตร์ และแนวคิดการเรียกซ้ำ (recursion) เบื้องต้น เช่น การคำนวณ factorial และ Fibonacci

03

การเขียนโปรแกรมแบบมอดูล

การแบ่งโปรแกรมออกเป็นฟังก์ชันย่อย ๆ เพื่อให้โค้ดอ่านง่าย บำรุงรักษาง่าย และนำกลับมาใช้ใหม่ได้ ซึ่งเป็นหลักการสำคัญของการเขียนโปรแกรมที่ดี



สัปดาห์ที่ 12 - 13

Numerical Computing with NumPy

สัปดาห์ที่ 12 – NumPy Arrays

การสร้างและใช้งาน array ใน NumPy การคำนวณแบบ vectorized ที่มีประสิทธิภาพสูงกว่า Python list ทั่วไป การดำเนินการทางคณิตศาสตร์บน array เช่น การบวก ลบ คูณ หาร และฟังก์ชันทางคณิตศาสตร์

สัปดาห์ที่ 13 – Matrix Operations

การสร้างและดำเนินการกับเมทริกซ์ การคูณเมทริกซ์ การหา transpose การดำเนินการเชิงเส้น (linear operations) และการประยุกต์ NumPy เพื่อแก้ปัญหาทางคณิตศาสตร์จริง เช่น การแก้ระบบสมการเชิงเส้น

สัปดาห์ที่ 14 - 15

บูรณาการ เวิร์กช็อป และทบทวนรายวิชา

สัปดาห์ที่ 14 – บูรณาการ / เวิร์กช็อปโครงงานย่อย

นักศึกษาฝึกแก้โจทย์ปัญหาทางคณิตศาสตร์ด้วย Python อย่างครบวงจร โดยบูรณาการความรู้ทุกหัวข้อที่เรียนมาตลอดภาคการศึกษา ตั้งแต่โครงสร้างพื้นฐานจนถึง NumPy

สัปดาห์ที่ 15 – ทบทวนรายวิชา + นำเสนอโครงงาน

ทบทวนเนื้อหาทั้งหมดของรายวิชา นักศึกษานำเสนอโครงงานย่อย และสรุปบทเรียนที่ได้รับตลอดภาคการศึกษา ก่อนเข้าสู่การสอบปลายภาค

ภาพรวมตารางสอนทั้งภาคการศึกษา

รายวิชานี้ครอบคลุม 15 สัปดาห์การเรียนการสอน แบ่งเป็น 4 ช่วงหลัก ได้แก่ พื้นฐาน Python, โครงสร้างการควบคุม, โครงสร้างข้อมูลและ NumPy และการบูรณาการ โดยมีการสอบกลางภาคและปลายภาคเป็นจุดวัดผลหลัก

การวัดและประเมินผล

สัดส่วนคะแนน

30%

สอบกลางภาค

30%

สอบปลายภาค

30%

งานมอบหมาย/โครงงาน/ทดสอบย่อย

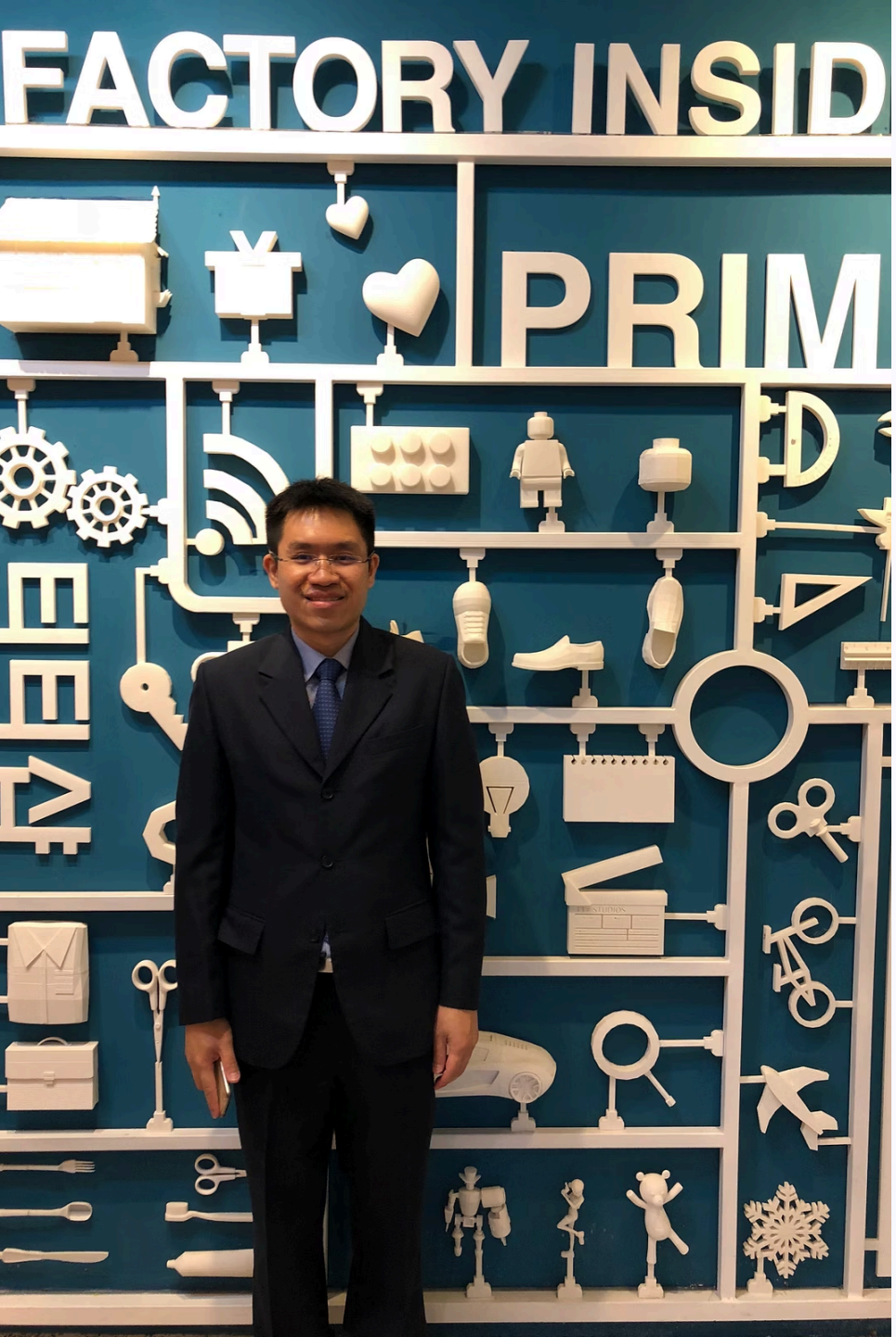
10%

การเข้าชั้นเรียนและการบ้าน

แนวทางการตัดเกรด

ใช้การตัดเกรดแบบ **อิงกลุ่มและอิงเกณฑ์** ร่วมกัน โดยพิจารณาจากค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (S.D.) ของกลุ่มนักศึกษาในแต่ละตอน

- นักศึกษาควรให้ความสำคัญกับงานมอบหมายและการเข้าชั้นเรียน เนื่องจากรวมกันคิดเป็น 40% ของคะแนนทั้งหมด



อาจารย์ผู้สอน – ตอนที่ 1

รศ. ดร.อภิชาต ศุรธณี

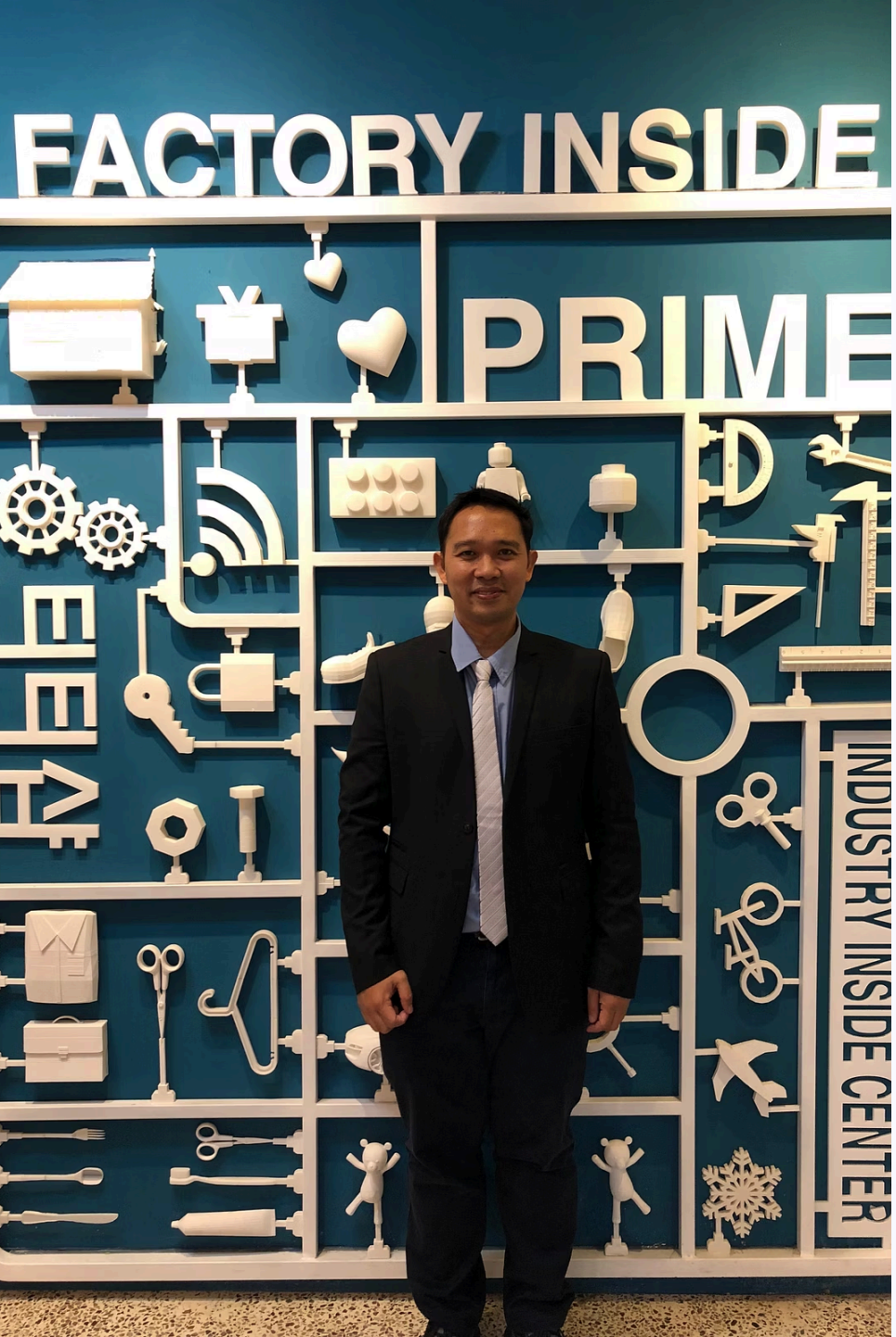
ตอนที่: 1

เวลาเรียน: วันอังคาร 13.00 - 17.00 น.

ห้องเรียน: 78-1008

เวลาที่นักศึกษาสามารถเข้าพบ

- วันอังคาร 09.00 - 12.00 น.
- วันศุกร์ 09.00 - 12.00 น.



อาจารย์ผู้สอน – ตอนที่ 2

รศ. ดร.อนุชิต จิตพัฒนกุล

ตอนที่: 2

เวลาเรียน: วันพุธ 13.00 - 17.00 น.

ห้องเรียน: 75-504

เวลาที่นักศึกษาสามารถเข้าพบ

- วันพุธ 09.00 - 12.00 น.
- วันพฤหัสบดี 13.00 - 16.00 น.

สรุปรายวิชา – พร้อมเริ่มต้นการเรียนรู้

🎯 เป้าหมาย

เขียนโปรแกรม Python เพื่อแก้ปัญหาทางคณิตศาสตร์ได้อย่างมีประสิทธิภาพ

📚 เนื้อหา

15 สัปดาห์ ครอบคลุมตั้งแต่พื้นฐาน Python จนถึงการคำนวณเชิงตัวเลขด้วย NumPy

📊 การวัดผล

สอบกลางภาค 30% + สอบปลายภาค 30% + งานมอบหมาย 30% + เข้าเรียน 10%

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ | ภาคการศึกษาที่ 1/2569