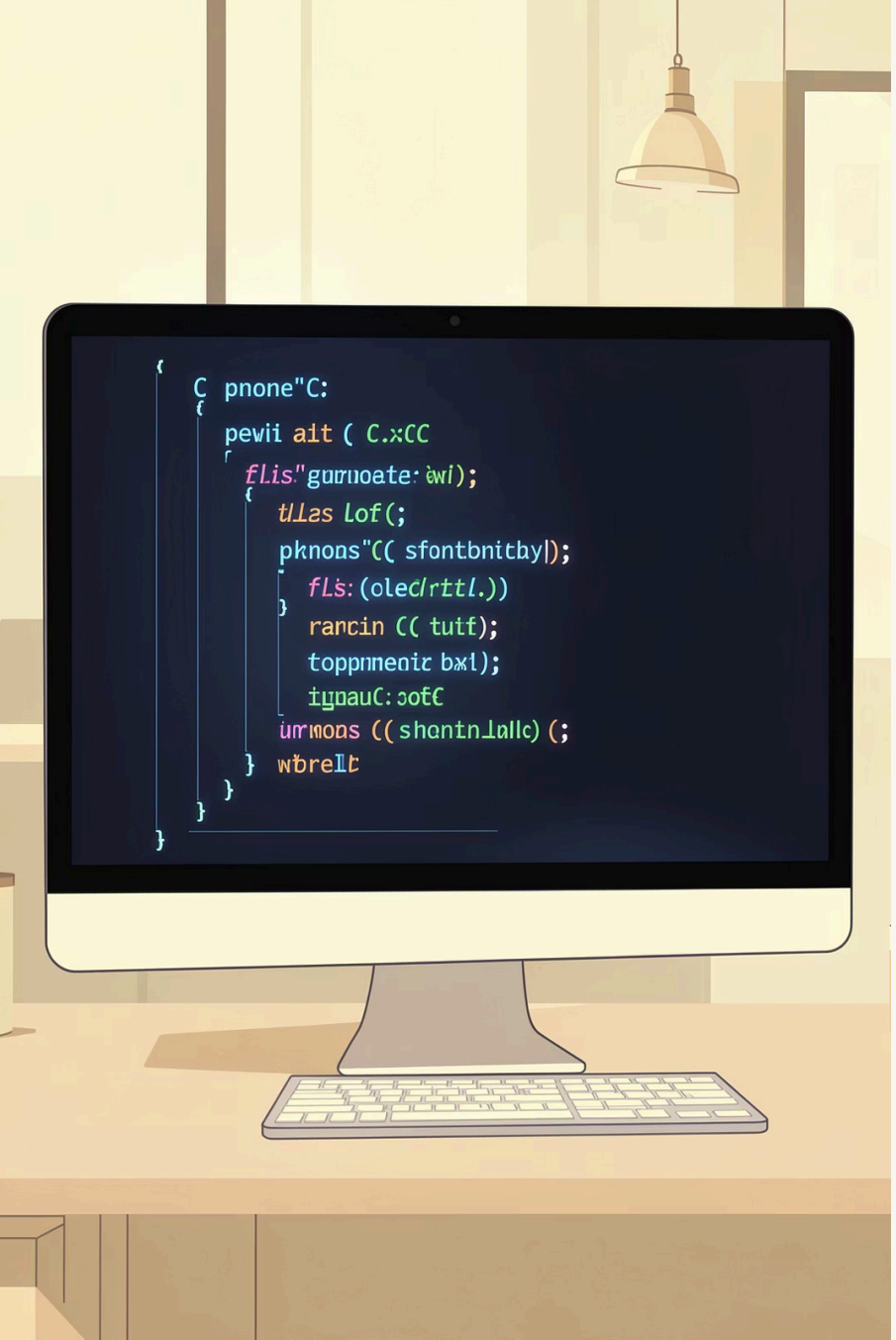


หลักการเขียนโปรแกรม

รายวิชา 20204-2004 | แผนการจัดการเรียนรู้ 18 สัปดาห์ ครอบคลุมเนื้อหาตั้งแต่พื้นฐานภาษาคอมพิวเตอร์ โครงสร้างการควบคุมโปรแกรม ไปจนถึงการพัฒนาซอฟต์แวร์อย่างมีระบบด้วยภาษาซี

ภาพรวมรายวิชา

โครงสร้างรายวิชา 18 สัปดาห์

สัปดาห์ที่ 1–4

พื้นฐานภาษาคอมพิวเตอร์ ผังงาน รหัสเทียม และขั้นตอนการพัฒนาซอฟต์แวร์

1

2

สัปดาห์ที่ 5–10

โครงสร้างภาษาซี ตัวแปร ชนิดข้อมูล ตัวดำเนินการ การรับ/แสดงผลข้อมูล

สัปดาห์ที่ 11–15

โครงสร้างควบคุม: การตัดสินใจ (if, switch) และการวนซ้ำ (for, while, do-while)

3

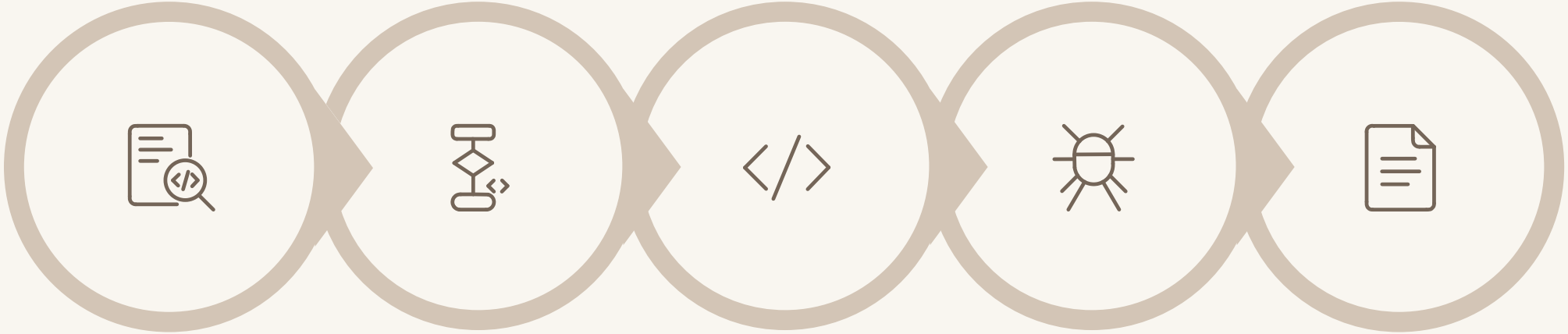
4

สัปดาห์ที่ 16–18

อาร์เรย์ ฟังก์ชัน การประยุกต์สร้างโปรแกรมจริง จริยธรรม และเศรษฐกิจพอเพียง

ขั้นตอนการพัฒนา

5 ขั้นตอนหลักในการพัฒนาโปรแกรม



วิเคราะห์ปัญหา

ออกแบบขั้นตอน

เขียนโปรแกรม

ทดสอบและแก้ไข

จัดทำเอกสาร

กระบวนการพัฒนาซอฟต์แวร์อย่างเป็นระบบช่วยให้โปรแกรมทำงานถูกต้อง มีประสิทธิภาพ และบำรุงรักษาได้ง่ายในระยะยาว

เครื่องมือออกแบบ

ผังงาน vs รหัสเทียม

ผังงาน (Flowchart)

ใช้สัญลักษณ์กราฟิก

เหมาะสำหรับนำเสนอภาพรวม

เหมาะกับผู้เริ่มต้น

อ่านง่ายแต่แก้ไขยาก

รหัสเทียม (Pseudo Code)

เขียนคล้ายภาษาอังกฤษผสมโครงสร้างโปรแกรม

กระชับ ยืดหยุ่น

แก้ไขง่าย รวดเร็ว

เหมาะใช้ก่อนเขียนโค้ดจริง

เลือกใช้ให้เหมาะกับสถานการณ์

ทั้งสองเครื่องมือช่วย**วางแผนก่อนเขียนโปรแกรม** ลดข้อผิดพลาดและประหยัดเวลาในการพัฒนา

- **ผังงาน** เหมาะสำหรับอธิบายให้ผู้อื่นเข้าใจภาพรวม
- **รหัสเทียม** เหมาะสำหรับร่างตรรกะก่อนลงมือเขียนโค้ด

MODULE 1

ภาษาซีและโครงสร้างพื้นฐาน

Preprocessor

#include <stdio.h>
เรียกใช้ไลบรารีมาตรฐาน Input/Output

ฟังก์ชัน main()

int main() จุดเริ่มต้นการทำงานของโปรแกรมทุกตัว

คำสั่งและตัวแปร

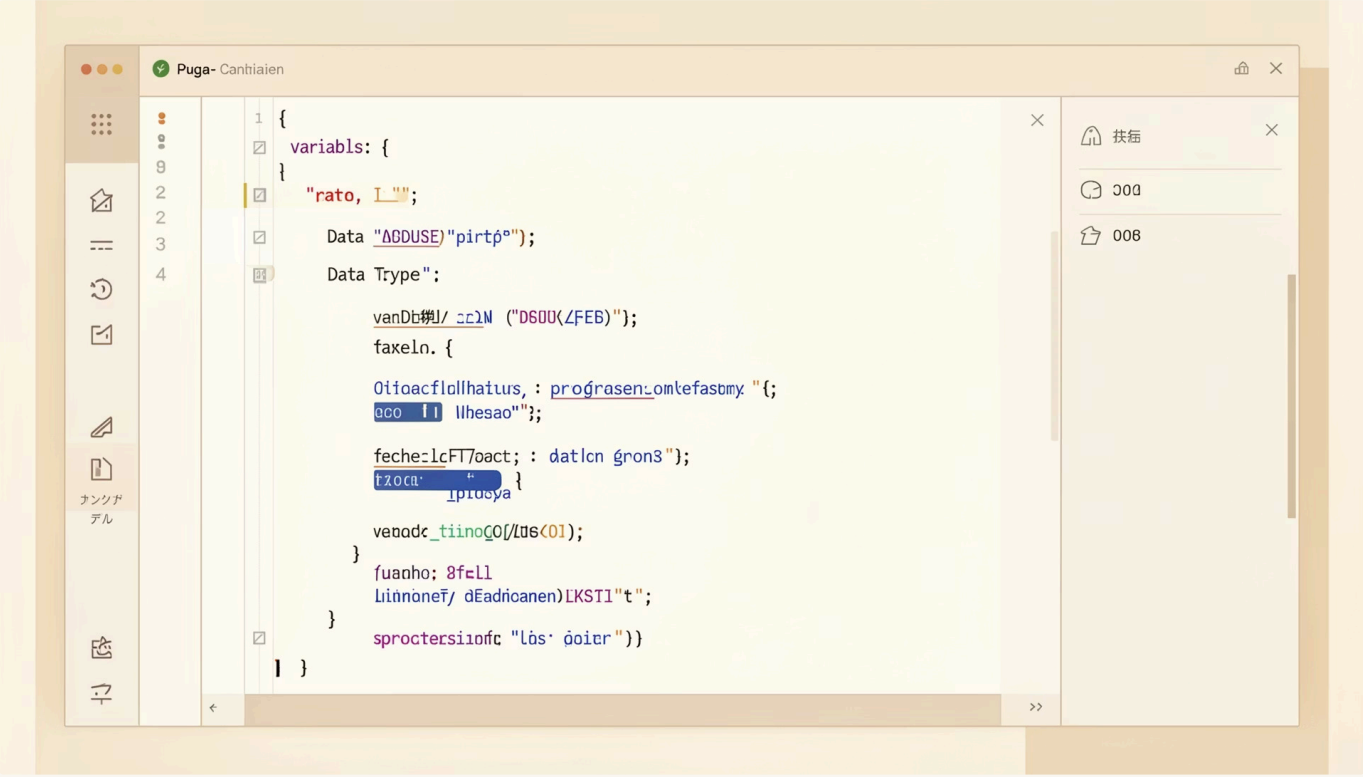
คำสั่งทำงานและตัวแปรท้องถิ่นอยู่ใน { }

การคืนค่า

return 0; บอกว่าโปรแกรมทำงานเสร็จสมบูรณ์

MODULE 2–4

ตัวแปร ชนิดข้อมูล และตัวดำเนินการ



ชนิดข้อมูลพื้นฐาน

- **int** — จำนวนเต็ม (เช่น 10, -5)
- **float / double** — ทศนิยม (เช่น 3.14)
- **char** — ตัวอักษรเดียว (เช่น 'A')

ตัวดำเนินการสำคัญ

- **Arithmetic:** +, -, *, /, %
- **Relational:** ==, !=, >, <, >=, <=
- **Logical:** && (AND), || (OR), ! (NOT)

กฎการตั้งชื่อ: ต้องขึ้นต้นด้วยตัวอักษรหรือ _ และห้ามซ้ำกับ Reserved Words

MODULE 3

การรับและแสดงผลข้อมูล

printf() — แสดงผล

ใช้ Format Specifiers ระบุชนิดข้อมูล

- %d สำหรับ int
- %f สำหรับ float / double
- %c สำหรับ char

scanf() — รับข้อมูล

ใช้เครื่องหมาย & (address-of) ระบุตำแหน่งตัวแปร

```
printf("Enter age: ");
scanf("%d", &age);
```

MODULE 5

การควบคุมทิศทางโปรแกรม

if / else

ตรวจสอบเงื่อนไขจริง/เท็จ รองรับ if-else หลายกรณีและ Nested if

switch / case

เลือกทำงานตามค่าคงที่ เหมาะสำหรับเมนูที่มีตัวเลือกจำนวนมาก

for Loop

วนซ้ำเมื่อทราบจำนวนรอบที่แน่นอนล่วงหน้า

while Loop

ตรวจสอบเงื่อนไขก่อน แล้วทำงานตราบเท่าที่เงื่อนไขเป็นจริง

do-while Loop

ทำงานอย่างน้อย 1 รอบเสมอ แล้วค่อยตรวจสอบเงื่อนไข

MODULE 6–7

อาร์เรย์ ฟังก์ชัน และการเขียนโปรแกรมที่ดี

Array (อาร์เรย์)

เก็บข้อมูลชนิดเดียวกันหลายค่าในชื่อเดียว ดัชนีเริ่มต้นที่ 0

```
int scores[5] = {80, 90, 75, 88, 95};
```

Function (ฟังก์ชัน)

แยกชุดคำสั่งเป็นบล็อกย่อย นำกลับมาใช้ใหม่ได้ มีการส่งค่า (Arguments) และการคืนค่า (Return Value)

แนวปฏิบัติที่ดี

- **Comment:** อธิบายโค้ดด้วย `//` หรือ `/* */`
- **Indentation:** ย่อหน้าโค้ดให้อ่านง่าย
- **Debugging:** แยกแยะ Syntax Error, Logic Error และ Runtime Error

สรุปและเป้าหมาย

สิ่งที่ผู้เรียนจะได้รับจากรายวิชานี้

🧠 ทักษะการคิดเชิงคำนวณ

วิเคราะห์ปัญหา ออกแบบขั้นตอนวิธีด้วยผังงานและรหัสเทียม

💻 ทักษะการเขียนโปรแกรม

เขียนโปรแกรมภาษาซีด้วย Dev-Cpp ตั้งแต่พื้นฐานจนถึงประยุกต์ใช้

⚖️ จริยธรรมและวิชาชีพ

ปลูกฝังเศรษฐกิจพอเพียงและจริยธรรมในการใช้เทคโนโลยีคอมพิวเตอร์