

Lab 06-1 Functions

กิจกรรมที่ 1 — ฝึกสร้างฟังก์ชันพื้นฐาน (Basic Function Creation)

เป้าหมาย: เข้าใจ Function Prototype, Function Definition และ Function Call

รายละเอียดงาน

ให้นักศึกษาเขียนโปรแกรมที่ประกอบด้วยฟังก์ชัน 3 ฟังก์ชันดังนี้

1. void printBanner(void)

- แสดงข้อความ

```
*****  
Welcome to Function Lab  
*****
```

2. int square(int n)

- คืนค่ากำลังสองของตัวเลข n

3. main()

- รับตัวเลขจากผู้ใช้ 1 ค่า
- เรียกใช้ฟังก์ชัน printBanner()
- เรียกใช้ฟังก์ชัน square() แล้วแสดงผลตัวเลขกำลังสอง

ผลลัพธ์

```
*****  
Welcome to Function Lab  
*****  
Enter a number: 5  
Square of 5 is 25
```

กิจกรรมที่ 2 — การส่งพารามิเตอร์และการคืนค่า (Parameter Passing & Return Value)

เป้าหมาย: ฝึกฟังก์ชันที่รับหลายพารามิเตอร์ มี return และเข้าใจการจับคู่ Actual/ Formal parameters

สถานการณ์

จากความรู้พื้นฐานเรื่อง “การคำนวณพื้นที่วงกลม” ให้นักศึกษาเขียนโปรแกรมคำนวณ

พื้นที่วงล้อ (Washer Area) = OuterCircleArea - InnerCircleArea

โดยใช้ฟังก์ชัน:

1. double circleArea(double r)
 - คืนค่าพื้นที่วงกลม (πr^2)
 - ให้ใช้ #define PI 3.14
2. double washerArea(double innerRadius, double outerRadius)
 - เรียกใช้ circleArea() ทั้งสองครั้ง
 - คืนค่าพื้นที่วงล้อ
3. main()
 - รับค่า inner และ outer จากผู้ใช้
 - เรียกใช้ washerArea()
 - แสดงผล

ผลลัพธ์

```
Enter inner radius: 5
Enter outer radius: 8
Washer area = 122.46
```

กิจกรรมที่ 3 — ระบบประเมินค่าจัดส่งพัสดุอัจฉริยะ (Smart Parcel Delivery Calculator)

สถานการณ์จำลอง: บริษัทขนส่งพัสดุแห่งหนึ่งต้องการโปรแกรมสำหรับคำนวณค่าจัดส่งพัสดุของลูกค้า โดยค่าจัดส่งจะขึ้นอยู่กับ "ระยะทาง" (คำนวณจากพิกัดต้นทางและปลายทาง) และ "น้ำหนัก" ของพัสดุ ให้นักศึกษาประยุกต์ใช้การสร้างฟังก์ชัน (User-defined Functions) ร่วมกับฟังก์ชันทางคณิตศาสตร์ `<math.h>` เพื่อพัฒนา ระบบนี้

รายละเอียดและเงื่อนไขการทำงานของโปรแกรม:

1. ส่วนของการคำนวณระยะทาง (Function 1)

- สร้างฟังก์ชันชื่อ `double calcDistance(double x1, double y1, double x2, double y2)`
- หน้าที: รับค่าพิกัดต้นทาง (x1, y1) และพิกัดปลายทาง (x2, y2) มาคำนวณหาระยะทางกระจัด (กิโลเมตร)
- คำใบ้: ให้ประยุกต์ใช้ทฤษฎีบทพีทาโกรัส $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$ โดยดึงฟังก์ชัน `sqrt()` (หารากที่สอง) และ `pow()` (ยกกำลัง) จากไลบรารี `<math.h>` มาใช้งาน จากนั้นให้คืนค่า (Return) ระยะทางที่คำนวณได้กลับไป

2. ส่วนของการคำนวณราคา (Function 2)

- สร้างฟังก์ชันชื่อ `double calcCost(double distance, double weight)`
- หน้าที: รับค่าระยะทาง (ที่ได้จาก Function 1) และ น้ำหนักพัสดุ (กิโลกรัม) มาประเมินราคาตามโครงสร้างดังนี้
 - ค่าบริการพื้นฐาน: 20 บาท
 - บวกเพิ่มตามระยะทาง: กิโลเมตรละ 5 บาท
 - บวกเพิ่มตามน้ำหนัก: กิโลกรัมละ 15 บาท
- ให้ฟังก์ชันคืนค่าผลลัพธ์ (Return) เป็นจำนวนเงินค่าจัดส่งสุทธิ

3. ส่วนของโปรแกรมหลัก main()

- ให้โปรแกรมใช้คำสั่งวนซ้ำ (Loop) เช่น `while` เพื่อรับข้อมูลพัสดุได้หลายๆ ชิ้นอย่างต่อเนื่อง โดยรับค่า พิกัดต้นทาง, พิกัดปลายทาง และ น้ำหนัก จากผู้ใช้งาน

- หากผู้ใช้งานป้อนค่าน้ำหนัก น้อยกว่าหรือเท่ากับ 0 ให้โปรแกรมหยุดการวนซ้ำและจบการทำงานทันที
- ภายใน Loop ให้ main() เรียกใช้ฟังก์ชัน calcDistance เพื่อหาระยะทางก่อน แล้วจึงนำระยะทางนั้นพร้อมกับน้ำหนักส่งเป็นอาร์กิวเมนต์ (Arguments) ไปให้ฟังก์ชัน calcCost ทำงาน
- แสดงผลลัพธ์ระยะทางและค่าจัดส่งของพัสดุแต่ละชิ้นออกทางหน้าจอให้สวยงาม

ผลลัพธ์

```
=== Parcel #1 ===
Enter start point (x1 y1): 2 2
Enter destination point (x2 y2): 7 9
Enter parcel weight (kg): 10
Distance: 8.60 km
Delivery Cost: 213.01 Baht
Press any key to continue or 0 to exit:

=== Parcel #2 ===
Enter start point (x1 y1): 5 6
Enter destination point (x2 y2): 9 9
Enter parcel weight (kg): 20
Distance: 5.00 km
Delivery Cost: 345.00 Baht
Press any key to continue or 0 to exit: 0

Program terminated.
```