ใบงานที่ 1: การแก้ปัญหาด้วยภาษาไพทอน (วงจร SDLC)

สถานการณ์ปัญหา (โจทย์):
"ร้านค้าสหกรณ์โรงเรียนต้องการโปรแกรมคำนวณราคาสินค้า โดยรับค่าราคาสินค้าและจำนวนที่ซื้อ จากนั้นแสดงยอดรวมทั้งหมด"

ส่วนที่ 1: การวิเคราะห์ความต้องการ (Requirement Analysis)

(ขั้นตอนนี้สำคัญที่สุด เพราะหากผิดพลาดจะทำให้ส่วนอื่นผิดตามและเสียเวลาแก้ไข)

หน้าที่ของโปรแกรม (Program Function): ....

ข้อมูลเข้า (Input): ....

ข้อมูลออก (Output): ....

ตัวแปรที่ต้องใช้: ....

ส่วนที่ 2: การออกแบบ (Design)

(ใช้การบรรยายเพื่อกำหนดการไหลของข้อมูลและโครงสร้างระบบให้ชัดเจนตามความต้องการที่วิเคราะห์ไว้)

1. บรรยายขั้นตอนการทำงานของระบบ (Algorithm): ให้นักเรียนเขียนบรรยายลำดับการทำงานของโปรแกรมเป็นข้อๆ โดยเรียงลำดับจาก เริ่มต้น -> การรับข้อมูล -> การคำนวณ -> การแสดงผล -> สิ้นสุด

คำตอบ: ....

(แนวทางคำตอบ: 1. เริ่มต้นโปรแกรม 2. รับค่าราคาสินค้าและจำนวน 3. นำราคาคูณกับจำนวนเพื่อหายอดรวม 4. แสดงยอดรวมที่คำนวณได้บนหน้าจอ 5. จบการทำงาน)

2. บรรยายการออกแบบหน้าจอผู้ใช้ (User Interface Design): อ้างอิงจากแหล่งข้อมูลที่ระบุว่าการออกแบบต้องกำหนด "หน้าจอผู้ใช้" ให้นักเรียนบรรยายว่าโปรแกรมจะ "พูดหรือถาม" อะไรกับผู้ใช้บ้างเพื่อให้ได้ข้อมูลที่ถูกต้อง

คำตอบ: ....

(แนวทางคำตอบ: โปรแกรมจะแสดงข้อความว่า "กรุณาใส่ราคาสินค้า:" และ "กรุณาใส่จำนวนที่ซื้อ:" จากนั้นจะแสดงผลว่า "ยอดรวมที่คุณต้องชำระคือ:")

ส่วนที่ 3: การป้องกันข้อผิดพลาด (Link to Testing)

(การตรวจสอบก่อนลงมือเขียนรหัสคำสั่ง (Coding) เพื่อลดข้อผิดพลาด)

การตรวจสอบตรรกะ (ป้องกัน Runtime Error): หากผู้ใช้ป้อนจำนวนสินค้าเป็น "ศูนย์" หรือ "ติดลบ" โปรแกรมจะทำงานได้หรือไม่? นักเรียนจะแก้ไขในผังงานอย่างไร?

คำตอบ:

....

การเตรียมเขียนรหัส (ป้องกัน Syntax Error): นักเรียนต้องระวังเรื่องไวยากรณ์ภาษาไพทอนในจุดใดบ้าง เพื่อไม่ให้เกิดข้อผิดพลาดตอนแปลภาษา (เช่น การสะกดคำสั่ง หรือตัวพิมพ์เล็ก-ใหญ่)

คำตอบ:

....

ส่วนที่ 4: การปรับปรุงอย่างต่อเนื่อง (Continuous Improvement)

(นำข้อเสนอแนะมาพัฒนาให้คอมพิวเตอร์ทำงานมีประสิทธิภาพยิ่งขึ้น)

ประเมินผล: หากต้องการให้โปรแกรมนี้ "เก่งขึ้น" (เช่น เพิ่มระบบสมาชิกหรือส่วนลด) นักเรียนอยากเพิ่มฟีเจอร์ใด?

คำตอบ:

....

LIVEWORKSHEETS