

แบบฝึกหัด 6

*****เขียนคำตอบด้วยลายมือ และส่งงานในรูปแบบไฟล์ pdf/jpg/png ตั้งชื่อไฟล์ตามรหัสனிสิต*****

ข้อ 1: ระบบพนักงานและเงินเดือนขั้นต่ำ

สถานการณ์: สมมติว่าบริษัทแห่งหนึ่งมีพนักงานทั้งแบบประจำ (Full-time) และพาร์ทไทม์ (Part-time) ซึ่งมีหลักเกณฑ์ในการคำนวณเงินเดือนแตกต่างกัน โดยมีรายละเอียดดังนี้

คำสั่ง:

1. ออกแบบคลาสพื้นฐานชื่อ Employee ซึ่งมีฟิลด์พื้นฐานเกี่ยวกับชื่อพนักงาน (protected String name) และเมธอด calculateSalary() (เป็น abstract method) สำหรับคำนวณเงินเดือนแบบ double และ printInfo() สำหรับแสดงข้อมูลพนักงาน ใน package ku.cs.swcon
2. สร้างคลาสย่อย 2 คลาส ซึ่งสืบทอดจาก Employee คือ
 - FullTimeEmployee
 - PartTimeEmployee
3. FullTimeEmployee ต้องมีฟิลด์เพิ่มเติม:
 - position (ตำแหน่ง): เช่น "Manager", "Developer", "Tester"
 - experienceYears (จำนวนปีที่มีประสบการณ์)
 - implement เมธอด calculateSalary() โดย เงินเดือนคำนวณจาก:

baseSalary = ตามตำแหน่ง

salary = baseSalary + (baseSalary * 0.03 * experienceYears)

ตำแหน่ง	baseSalary
Manager	50,000
Developer	35,000
Tester	30,000
อื่น ๆ	25,000

4. PartTimeEmployee ต้องมีฟิลด์เพิ่มเติม:

- hourlyRate (ค่าจ้างต่อชั่วโมง)
 - hoursWorked (จำนวนชั่วโมงที่ทำงาน)
 - เงินเดือน = hourlyRate * hoursWorked
5. ใช้คลาส EmployeeTest ที่ปรากฏในตัวอย่าง เพื่อทดสอบการทำงาน โดยสร้างพนักงานแต่ละประเภท และแสดงผลลัพธ์จาก calculateSalary() และ printInfo() ของ object จากคลาส FullTimeEmployee และ PartTimeEmployee

ตัวอย่าง Input:

```
package ku.cs.swcon;

public class EmployeeTest {

    public static void main(String[] args) {

        Employee emp1 = new FullTimeEmployee("Alice", "Manager", 5);
        Employee emp2 = new PartTimeEmployee("Bob", 200.0, 20);
        Employee emp3 = new FullTimeEmployee("Charlie", "Developer", 3);

        Employee[] employees = {emp1, emp2, emp3};

        for (Employee e : employees) {
            e.printInfo();
        }
    }
}
```

ตัวอย่าง Output:

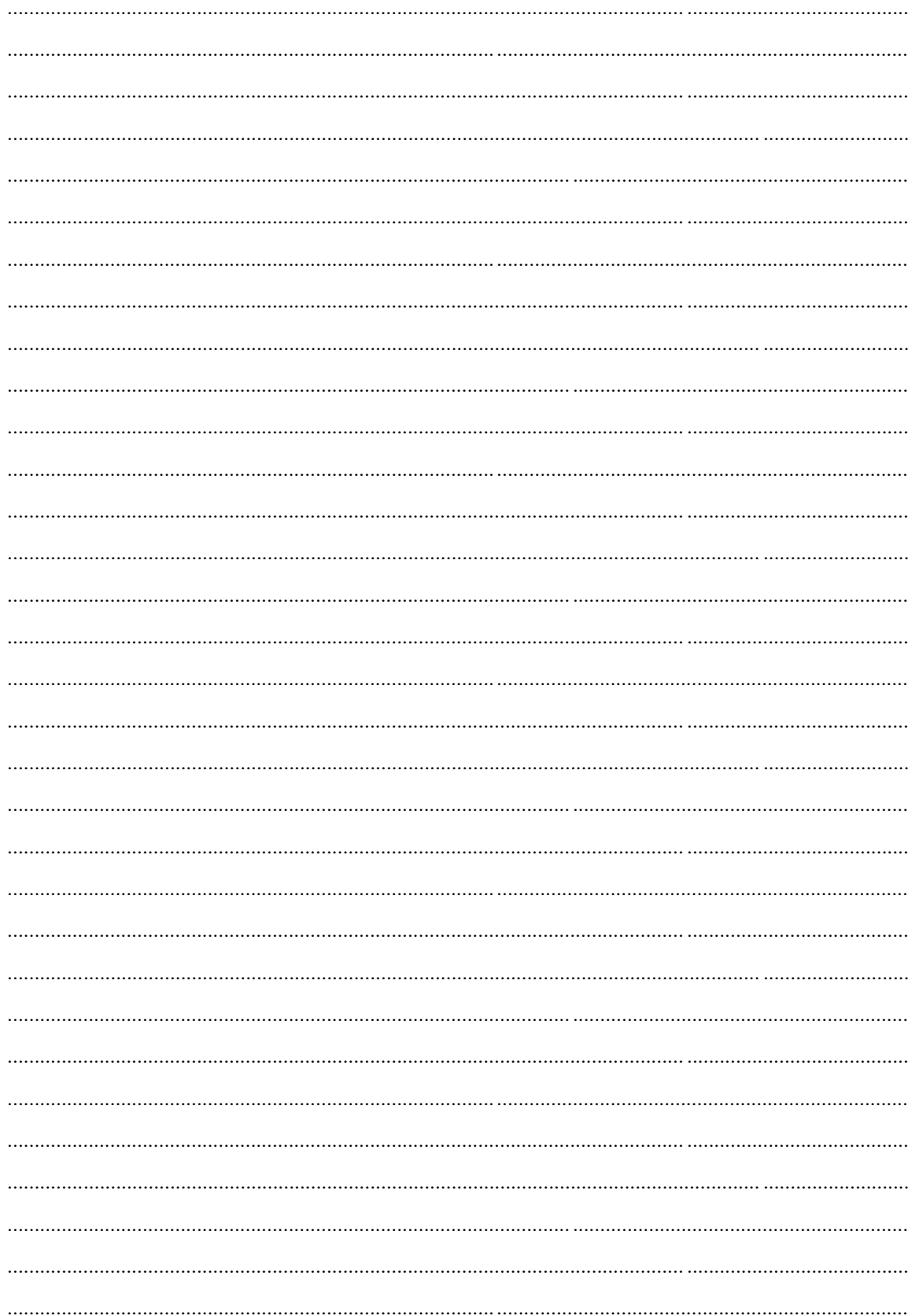
Full-Time Employee: Alice, Position: Manager, Experience: 5 years
Salary: 57500.0

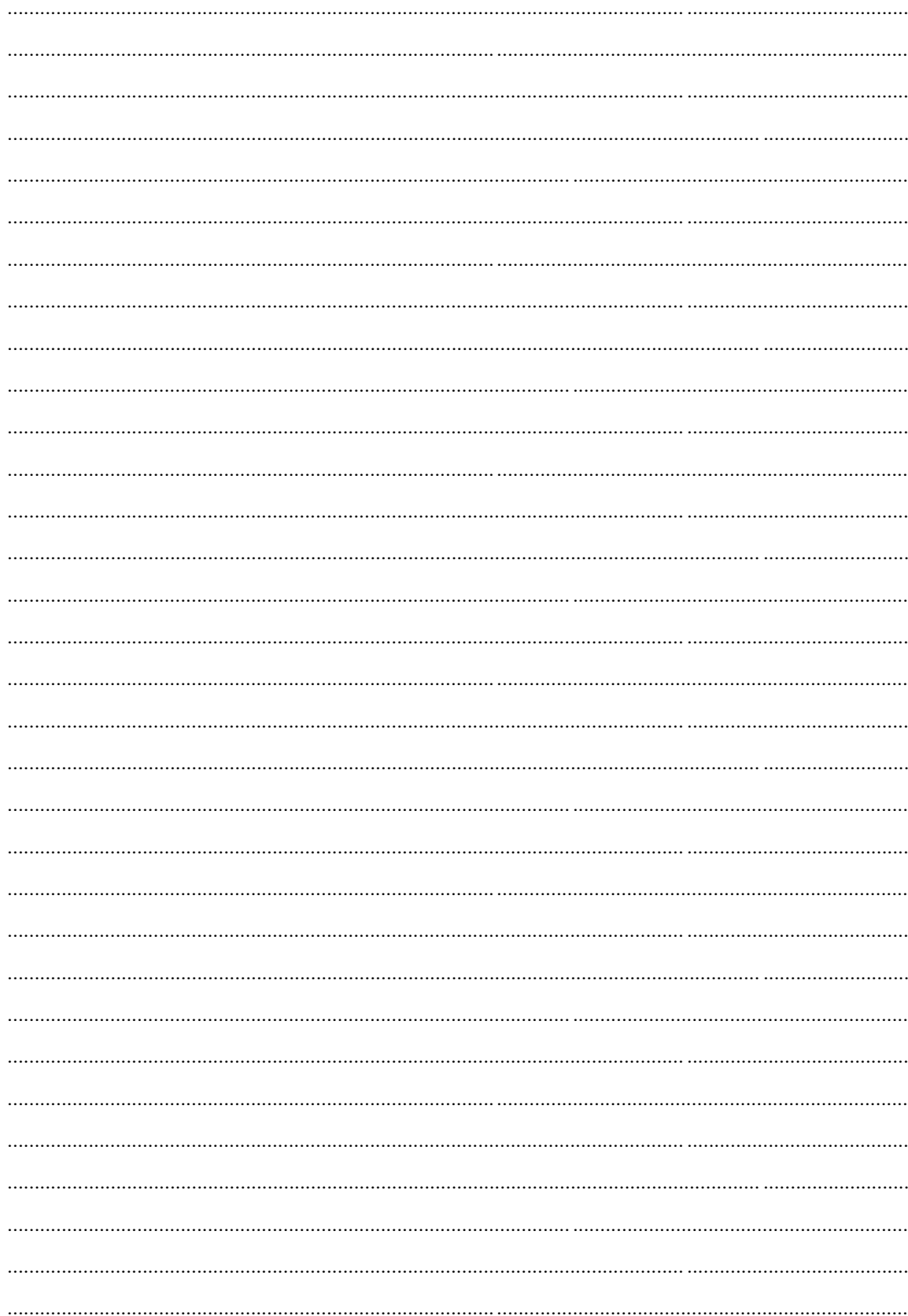
Part-Time Employee: Bob
Salary: 4000.0

Full-Time Employee: Charlie, Position: Developer, Experience: 3 years

Salary: 38150.0

[illegible]





ข้อ 2: ระบบจัดการหลักสูตรในมหาวิทยาลัย (University Course System)

สถานการณ์: ออกแบบระบบสำหรับเพิ่มรายวิชา ซึ่งแต่ละรายวิชาในมหาวิทยาลัยจะมีลักษณะแตกต่างกัน เช่น วิชาทั่วไป (GeneralCourse) และวิชาปฏิบัติการ (LabCourse) จะมีการจัดการการเรียนการสอนและจำนวนหน่วยกิตที่ต่างกัน ในการสร้างวิชานั้น รหัสวิชา (courseCode) ห้ามเปลี่ยนแปลงหลังจากตั้งแล้ว และระบบจะนับจำนวนรายวิชาทั้งหมดที่ถูกสร้าง (แบบ static counter)

คำสั่ง:

- สร้าง abstract class Course ใน package ku.cs.swcon
 - 필드 private final String courseCode
 - 필드 private String title
 - public static int COURSE_COUNT เพื่อเก็บจำนวนคลาสที่ถูกสร้าง
 - Constructor เพิ่มค่า courseCount ทุกครั้งที่มีการสร้างวิชา
 - abstract method getCreditHour() ที่ subclass ต้อง override
- สร้างคลาส GeneralCourse ที่ override getCreditHour() และ return credit = 3 พร้อมทั้งมี constructor ที่เหมาะสม
- สร้างคลาส LabCourse ที่ override getCreditHour() และ return credit = 1 พร้อมทั้งมี constructor ที่เหมาะสม
- ใช้คลาส CourseTester ดังตัวอย่างเพื่อแสดงจำนวนวิชาทั้งหมด และรายละเอียดของแต่ละวิชา

ตัวอย่าง Input:

```
package ku.cs.swcon;

public class CourseTester {

    public static void main(String[] args) {

        Course c1 = new GeneralCourse("GE101", "English Communication");
        Course c2 = new GeneralCourse("GE102", "Thai Civilization");
        Course c3 = new LabCourse("CS103L", "Programming Lab");

        System.out.println(c1.getInfo());
        System.out.println(c2.getInfo());
        System.out.println(c3.getInfo());
    }
}
```

```
System.out.println("จำนวนวิชาทั้งหมดที่เปิดสอน: " + Course.courseCount);
}
}
```

Output (ตัวอย่าง):

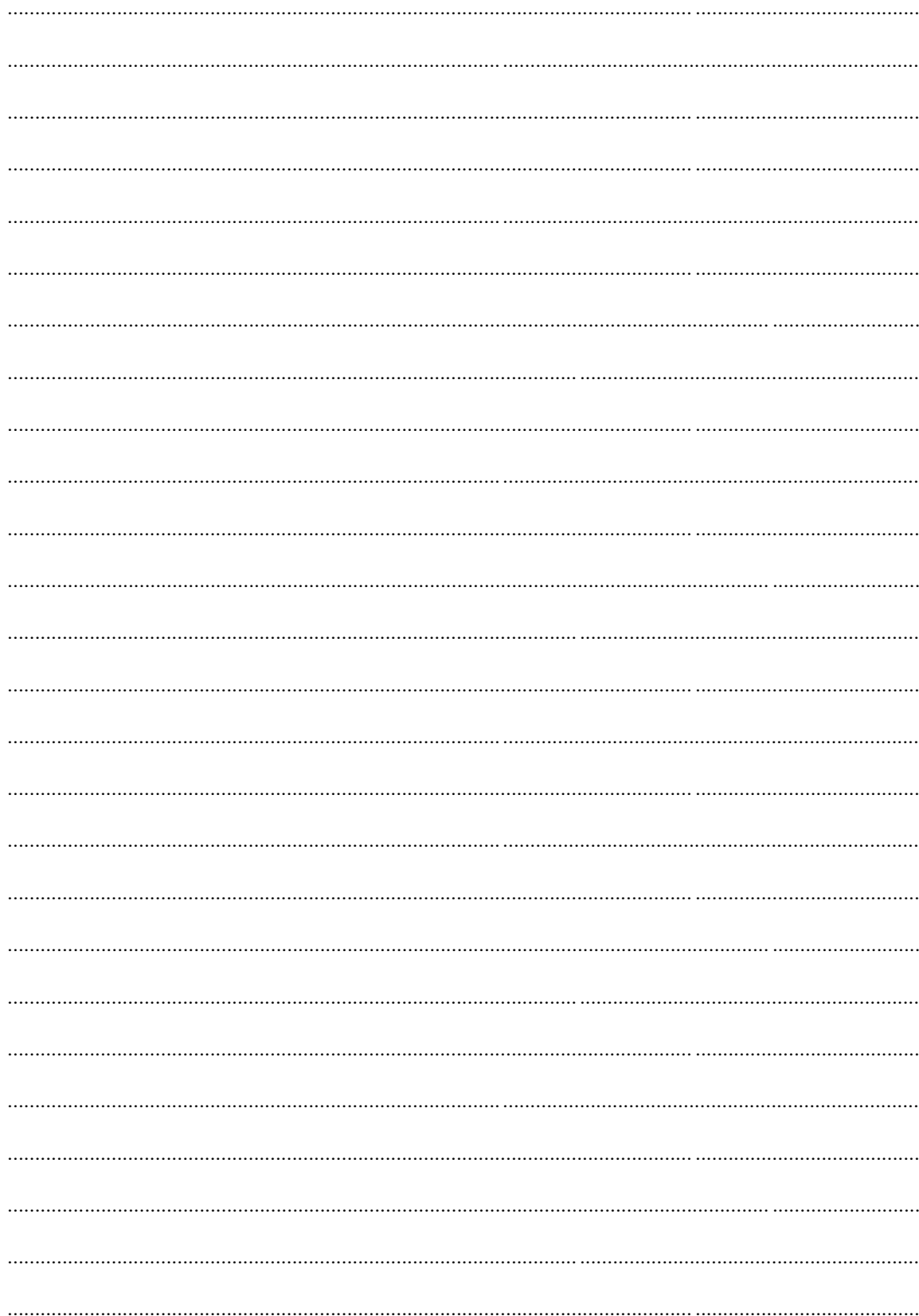
GE101 - English Communication (3 หน่วยกิต)

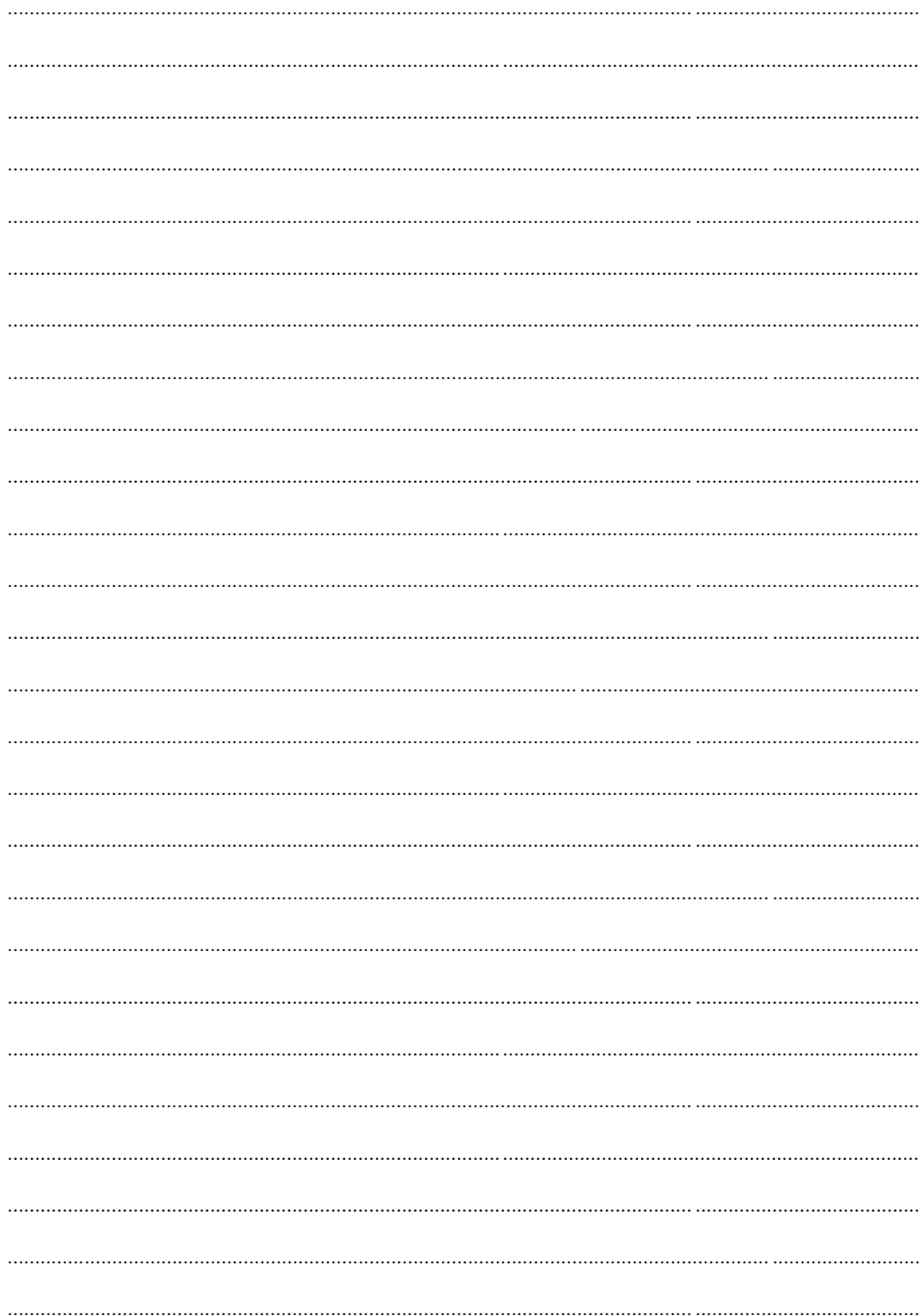
GE102 - Thai Civilization (3 หน่วยกิต)

CS103L - Programming Lab (1 หน่วยกิต)

จำนวนวิชาทั้งหมดที่เปิดสอน: 3

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.





ข้อ 3: ระบบติดตามการขนส่งพัสดุ (Package Delivery System)

สถานการณ์: พัสดุมีได้หลายประเภท เช่น พัสดุด่วน (ExpressPackage) และพัสดุธรรมดา (StandardPackage) ซึ่งมีการคำนวณค่าจัดส่งต่างกัน

คำสั่ง:

- สร้างคลาส Package ใน package ku.cs.swcon โดยมี:
 - private double weight (หน่วยเป็นกิโลกรัม)
 - private String destination
 - เมธอด calculateShippingFee() return 0.0
- สร้างคลาส ExpressPackage ที่สืบทอดจาก Package ทำการ override calculateShippingFee() โดยคิดค่าจัดส่งที่:
 - 100 บาทต่อกิโลกรัม + ค่าบริการพิเศษ 50 บาท
- สร้างคลาส StandardPackage ที่สืบทอดจาก Package ทำการ override calculateShippingFee() โดยคิดค่าจัดส่ง:
 - 50 บาทต่อกิโลกรัม (ไม่มีค่าบริการเพิ่มเติม)
- ใช้คลาส DeliveryTest ดังตัวอย่างเพื่อสร้างพัสดุทั้งสองประเภท แสดงจุดหมายปลายทาง น้ำหนัก และค่าจัดส่งที่คำนวณได้

ตัวอย่าง Input:

package ku.cs.swcon;

```
public class DeliveryTest {  
    public static void main(String[] args) {  
        Package express = new ExpressPackage(2.5, "เชียงใหม่");  
        Package standard = new StandardPackage(3.0, "กรุงเทพฯ");  
  
        System.out.println("ปลายทาง: " + express.getDestination() + " ค่าจัดส่ง: " +  
            express.calculateShippingFee());  
        System.out.println("ปลายทาง: " + standard.getDestination() + " ค่าจัดส่ง: " +  
            standard.calculateShippingFee());  
    }  
}
```

}

Output (ตัวอย่าง):

ปลายทาง: เชียงใหม่ ค่าจัดส่ง: 300.0

ปลายทาง: กรุงเทพฯ ค่าจัดส่ง: 150.0

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

