

**Name-Surname: Phuriphat Hemakul****ID Number: 6752300313****Section:**

1. **คำสั่ง:** ทำการสร้าง DisplayName.java โดยมีรายละเอียดดังต่อไปนี้ ให้ทำการเปลี่ยน "YOURNAME" เพื่อแสดงรหัสนักศึกษา ชื่อ และ นามสกุล เช่น "6722300149 Chanakarn KINGKAEW" โดยทำการเขียนผ่าน method: display() โปรดทำโค้ดที่เหลือให้สามารถแสดงค่าออกมาได้ **[L3-1 DisplayName.java]**

```
class Name {  
    public void display() {  
  
        System.out.println("67XXXXXX YOUR NAME HERE");  
    }  
}  
  
public class DisplayName {  
    public static void main(String[] args) {  
        // เขียนโค้ดตรงนี้ ส่งไฟล์ java [L3-1 DisplayName.java]  
  
    }  
}
```

2. **คำสั่ง:** ให้เขียน method ชื่อ calculateAverage(...) ที่รับค่า array ของ int และ ส่งค่าเฉลี่ยของ array กลับมา จากนั้นให้แสดง output **[L3-2 Calculate.java]**

เช่น กำหนดให้ int [ ] arr = {1,2,3,4,5,6,7,8,9,10};

Input	Output
{1,2,3,4,5,6,7,8,9,10}	Average: 5.5

Code เริ่มต้น:

```
class Calculate {  
  
    public _____ calculateAverage (_____ ) {  
  
  
        return _____;  
    }  
  
}  
  
public class CalculateTest {  
  
    public static void main (String [] args) {  
  
        int [ ] arr = {1,2,3,4,5,6,7,8,9,10};  
  
    }  
  
}
```

นอกจากนั้น ให้เพิ่ม method และ แสดงการใช้งานและผลลัพธ์ผลใน Main Class

- **int add(int a, int b)** ทำการบวกเลข a กับ b
- **int subtract(int a, int b)** ทำการลบเลข a กับ b
- **int multiply(int a, int b)** ทำการคูณเลข a กับ b
- **double divide(int a, int b)** ทำการหารเลข a กับ b โดยมีเงื่อนไข → return -1 ถ้า b == 0 (หาค่าไม่ได้ (undefined))

3. **คำสั่ง:** ให้นักศึกษาทำการสร้างคลาส Account (บัญชี) [L3-3 Account.java]

3.1 โดยกำหนดให้มีการเก็บข้อมูลเลขที่บัญชี (account number) ชื่อ-นามสกุล (name) และ ยอดเงิน (balance)

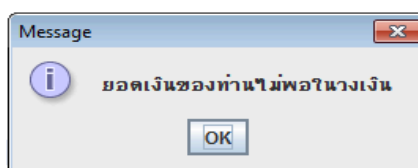
3.2 ให้แก้ไขโค้ดให้สามารถใช้ method withdraw() ถอนเงิน เงื่อนไขคือ ไม่สามารถถอนเงินได้เมื่อ ยอดเงิน balance ถ้าเกินให้ใช้ Message Dialog มาช่วยในการเตือน, เก็บรายละเอียดให้ครอบคลุม, เขียนคอมเมนต์

ใช้ Concept: Encapsulation, Control Logic

```
class Account {  
    /// ทำเพิ่มเติมเพื่อเก็บข้อมูล  
    private double balance;  
    public Account(double initialBalance) {  
        if (initialBalance > 0.0) balance=initialBalance;  
    }  
    public void deposit(double amount){ /// การเพิ่มเงินเข้าบัญชี  
        balance=balance+amount;  
    }  
    public void withdraw(double amount){ /// การถอนเงินออกบัญชี  
  
        balance=balance-amount;  
    }  
    public double getBalance(){ /// ดูยอดเงินในบัญชี  
        return balance;  
    }  
}  
  
public class AccountTest {  
    public static void main (String [] args) {  
        Account me = new Account(100); //เงิน 100  
        me.deposit(300); // เงิน 100+300 = 400  
        System.out.println(me.getBalance());  
        me.withdraw(500); // เกินวงเงิน  
        System.out.println(me.getBalance()); }  
}
```

Output

400



400

4. เมื่อเสร็จเขียน **Class Diagram** (Account.java) ของโค้ดที่เสร็จแล้วด้วย โดยสามารถใช้โปรแกรมต่าง ๆ ได้ไม่จำกัด

แนะนำ <https://app.diagrams.net/>

