



Lab 6 Solution

◆ 1. Superclass: Calculator

✓ المطلوب

- Method باسم `compute(double a, double b)`
- ترجع مجموع الرقمين

```
public class Calculator {  
  
    public double compute(double a, double b) {  
        return a + b;  
    }  
}
```

◆ 2. Subclass: AdvancedCalculator

✓ المطلوب

- extends باستخدام Calculator ترث من
- **Override** للـ `compute(double a, double b)` ليعيد الضرب
- **Overload** للـ `compute()` بنسختين:
 - `compute(double a)` → مربع العدد
 - `compute(double a, double b, double c)` → مجموع 3 أعداد

```
public class AdvancedCalculator extends Calculator {  
  
    // Method Overriding  
    @Override  
    public double compute(double a, double b) {  
        return a * b;  
    }  
  
    // Method Overloading  
    public double compute(double a) {  
        return a * a;  
    }  
  
    public double compute(double a, double b, double c) {  
        return a + b + c;  
    }  
}
```

◆ 3. Main Class

✓ المطلوب

- إنشاء كائن من Calculator
- إنشاء كائن من AdvancedCalculator
- إلبان الفرق بين compute () استدعاء كل نسخ:
 - o **Overriding**
 - o **Overloading**

```
public class Main {
    public static void main(String[] args) {

        Calculator calc = new Calculator();
        AdvancedCalculator advCalc = new AdvancedCalculator();

        // From Calculator
        System.out.println("Calculator (sum): " +
            calc.compute(5, 3));

        // Overriding
        System.out.println("AdvancedCalculator (product): " +
            advCalc.compute(5, 3));

        // Overloading
        System.out.println("Square: " + advCalc.compute(4));
        System.out.println("Sum of three: " + advCalc.compute(1,
            2, 3));
    }
}
```



Expected Output

```
Calculator (sum): 8.0
AdvancedCalculator (product): 15.0
Square: 16.0
Sum of three: 6.0
```



ملخص سريع

- extends → وراثه
 - @Override → موروثة method تغيير سلوك
 - Overloading → نفس الاسم + معاملات مختلفة
-