**ให้นักศึกษา สร้าง class หาร สามารถใช้ บวก และ ลบ คูณ จาก class ก่อนหน้านี้ได้**

|  |
| --- |
| Code C++  #include <iostream>  using namespace std;  class Calculator {  public:  int add(int a, int b) {  return a + b;  }  };  class SubtractionCalculator : public Calculator {  public:  int subtract(int a, int b) {  return a - b;  }  };  class MultiplicationCalculator : public SubtractionCalculator {  public:  int multiply(int a, int b) {  return a \* b;  }  };  class DivisionCalculator : public MultiplicationCalculator {  public:  int divide(int a, int b) {  return a / b;  }  };  int main() {  DivisionCalculator mulCalc;  cout << "Addition (10 + 5): " << mulCalc.add(10, 5) << endl;  cout << "Subtraction (10 - 5): " << mulCalc.subtract(10, 5) << endl;  cout << "Multiplication (10 \* 5): " << mulCalc.multiply(10, 5) << endl;  cout << "Division (10 / 5): " << mulCalc.divide(10, 5) << endl;  return 0;  } |

ผลการทดลอง

|  |
| --- |
|  |

|  |
| --- |
| Code Python  class Calculator:  def add(self, a, b):  return a + b  class SubtractionCalculator(Calculator):  def subtract(self, a, b):  return a - b  class MultiplicationCalculator(SubtractionCalculator):  def multiply(self, a, b):  return a \* b  class DivisionCalculator(MultiplicationCalculator):  def divide(self, a, b):  return a / b  if \_\_name\_\_ == "\_\_main\_\_":  mul\_calc = DivisionCalculator()  print(f"Addition (10 + 5): {mul\_calc.add(10, 5)}")  print(f"Subtraction (10 - 5): {mul\_calc.subtract(10, 5)}")  print(f"Multiplication (10 \* 5): {mul\_calc.multiply(10, 5)}")  print(f"Division (10 / 5): {mul\_calc.divide(10, 5)}") |

ผลการทดลอง

|  |
| --- |
|  |

ให้นักศึกษา เลือกอุปกรณ์ไฟฟ้า 3 อุปกรณ์

|  |
| --- |
| **1. การออกแบบคลาสลูก \_\_\_\_\_\_\_\_\_\_**  **คลาส \_\_\_\_\_\_\_:** เป็นตัวแทนของ\_\_\_\_\_\_\_ที่สามารถเปิด/ปิดได้เหมือนอุปกรณ์ทั่วไป แต่มีฟังก์ชันเพิ่มเติม:   * **คุณสมบัติใหม่:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * **พฤติกรรมเพิ่มเติม:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |
| --- |
| **2. การออกแบบคลาสลูก \_\_\_\_\_\_\_\_\_\_**  **คลาส \_\_\_\_\_\_\_:** เป็นตัวแทนของ\_\_\_\_\_\_\_ที่สามารถเปิด/ปิดได้เหมือนอุปกรณ์ทั่วไป แต่มีฟังก์ชันเพิ่มเติม:   * **คุณสมบัติใหม่:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * **พฤติกรรมเพิ่มเติม:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |
| --- |
| **3. การออกแบบคลาสลูก \_\_\_\_\_\_\_\_\_\_**  **คลาส \_\_\_\_\_\_\_:** เป็นตัวแทนของ\_\_\_\_\_\_\_ที่สามารถเปิด/ปิดได้เหมือนอุปกรณ์ทั่วไป แต่มีฟังก์ชันเพิ่มเติม:   * **คุณสมบัติใหม่:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * **พฤติกรรมเพิ่มเติม:**   + \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Code

|  |
| --- |
|  |

ผลการทดลอง

|  |
| --- |
|  |

|  |
| --- |
| Code Python |

ผลการทดลอง

|  |
| --- |
|  |