

این کد یک ماشین حساب ساده را پیاده‌سازی می‌کند. در زیر توضیحی کامل درباره هر خط کد آورده شده است:

```
#include <iostream>
```

این دستور، کتابخانه <iostream> را وارد می‌کند که امکان ورود و خروج از استاندارد ورودی/خروجی را فراهم می‌کند.

```
class Calculator {
public:
    int Add(int a, int b) {
        return a + b;
    }

    int Subtract(int a, int b) {
        return a - b;
    }

    void RunTests() {
        int addResult = Add(2, 3);
        CheckResult("Add Test", addResult, 5);

        int subtractResult = Subtract(5, 3);
        CheckResult("Subtract Test", subtractResult, 2);
    }
```

```
private:
```

```
    void CheckResult(const std::string& testName, int result, int expectedResult) {
        std::cout << "Test: " << testName << std::endl;
        std::cout << "Result: " << result << std::endl;
        std::cout << "Expected Result: " << expectedResult << std::endl;
        std::cout << "Test " << (result == expectedResult ? "Passed" : "Failed") << std::endl;
        std::cout << std::endl;
    }
};
```

این بخش، تعریف کلاس "Calculator" را شامل می‌شود. این کلاس دارای دو عملیات جمع (Add) و کاهش (Subtract) است که به ترتیب دو عدد صحیح را به عنوان ورودی می‌گیرد و نتیجه‌ی آن عملیات را برمی‌گرداند. همچنین تابع RunTests برای اجرای تست‌های واحد ایجاد شده است. تست جمع ابتدا عملیات جمع دو عدد را با ورودی‌های 2 و 3 انجام می‌دهد و نتیجه را در متغیر addResult ذخیره می‌کند. سپس با استفاده از تابع CheckResult، نتیجه‌ی عملیات جمع را با مقدار مورد انتظار (5) بررسی می‌کند. تست کاهش نیز به همین صورت عمل می‌کند.

```
int main() {
    Calculator calculator;
    calculator.RunTests();

    return 0;
}
```

در این بخش، تابع main() تعریف شده است. در ابتدا یک شیء از کلاس Calculator با نام calculator ایجاد می‌شود. سپس تابع RunTests این شیء را فراخوانی می‌کند تا تست‌های واحد انجام شود. در انتها

مقدار صفر به عنوان خروجی تابع main برگردانده می‌شود. این بدان معناست که برنامه بدون مشکل اجرا شده است.