

[illegible]

API 設計 ( 40% )		
為 Decimal、Integer、Complex 重載 operator <<, >>, +, -, *, / , 使這三種型別可以進行複合運算，並且可以使用 std::cout 以及 std::cin 進行輸出輸入。	Integer i; Decimal d; Complex c; cin >> i >> d >> c; cout << i + d; cout << i + c; cout << d + c;	20%
為 Decimal、Integer、Complex 重載 Constructor，可以直接賦予字串進行初始化， <b>其中字串內容可以是任意運算式</b> 。	Integer i = "12345"; Decimal d = "0.3 * 3"; Complex c = "1.0+(5*2)i";	10%
繼承共同基底類別或者使用其它設計技巧，使得三種型別可以放入同一容器中，並且可以走訪所有儲存的元素然後加以輸出。	Integer i = "123"; Decimal d = "123.3"; Complex c = "123+123i";  vector<NumberObject*> nums; nums.push_back(&i); nums.push_back(&d); nums.push_back(&c); for(const auto& num : nums) cout << *num << endl;	10%
使用者體驗 ( 0 ~ 25% )		
防呆測試，一個測試 2 分。		0~10%
呈現方式(美化 Console 或者以 C++開發 GUI)。		0~15%

- 有些功能細分為可帶入變數或者只能直接輸入 Constant 來計分。
- 前半部的說明可參考先前的文件。
- 超過 100 分以 100 分計算。
- 紅色標記為上一份文件沒有明確指出的部分。
- 藍色標記原先為必要項目，這次被加入計分項目中。
- 原先的變數功能分數已被拆散至其它項目中。