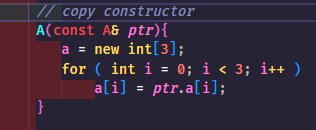
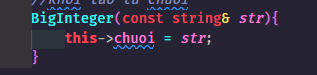
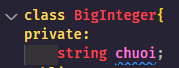
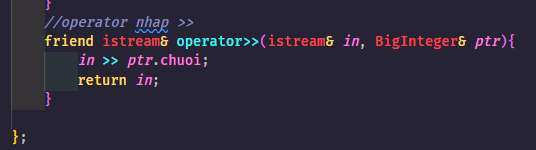
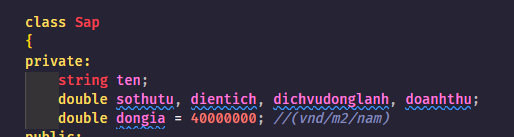
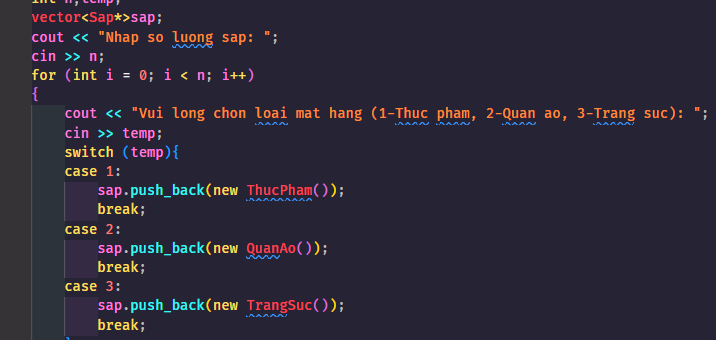
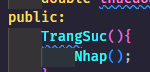
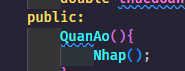
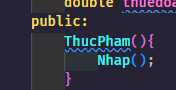
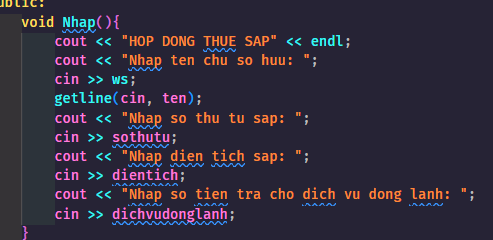
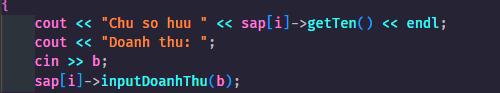
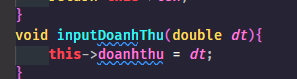
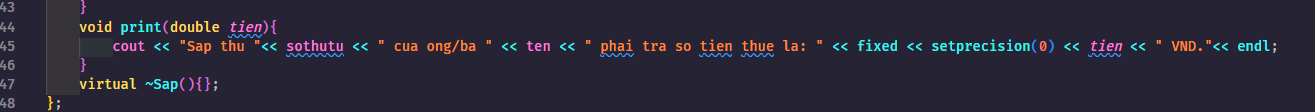
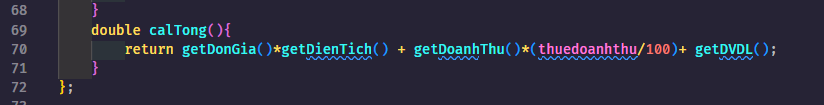
Câu 1:

1. Output: “3 4”. Hàm doSomething yêu cầu truyền 2 parameter a và b,  
    a thuộc về class A và b thuộc về class B. Nhưng ở hàm main, 2 parameter đó được khai báo thuộc về class B, mà lại có thể sử dụng được.   
   Giải thích vấn đề này là do đa hình, lớp dẫn xuất B có thể dùng cho các tham chiếu A.  
    Tuy nhiên, khi parameter đầu tiên truyền vào (là x), và gọi hàm f(3) từ a nó lại chạy hàm f() của class A vì trong hàm doSomething ta truyền a vào không có tham chiếu. Nếu đổi doSomething(A& a, B& b) thì sẽ khác. Về phần b thì cùng kiểu với ptr đã khai, nên không có gì thay đổi.
2. Vấn đề ở việc quản lý bộ nhớ.   
     
   Giả sử khi gán a = b trong hàm init, khi không có hàm sao chép(copy constructor), máy tính sẽ tự tạo một trình sao chép mặc định và chép tất cả các thành phần dữ liệu bao gồm cả con trỏ. Điều này dẫn đến con trỏ a và b đều trỏ cùng vào một vùng nhớ, gay ra lỗi.   
   fix thêm một cái con constructor copy vào   
     
     
   trong file 2015C1b.cpp  
     
   1c)   
   tại sao mình bỏ con Singleton vào hàm private : không được gọi ở đâu khác ngoài cái hàm private nó chỉ đước ở trong class thôi  
   để nó tạo ra đúng một thầy instance không có thằng thứ 2  
   muốn chắc chắn hơn nữa thì gán 1 cái operator = copy contructor và một cái operator bằng, để người ta không thể sao chép hay gán inst vào bất kì nào đó  
     
     
   câu 2:   
   khởi tạo từ chuỗi:  
      
   constructor truyền vào nó một cái tham chiếu string và gán cái chuối   
     
   khỏi tạo mặc định   
    có thể viếc cách rõ hơn   
   BigInterger(){  
    this-> chuoi=”0”; mà chuyên nghiệp hơn người ta sẽ viết kiểu kia.

mình viết nhập   
  
kiểu so sánh bằng:  
khi viết một cái operator so sánh thì nên dùng kiểu bool và có const, để đảm bảo tính toàn vẹn của dữ liệu ngoài ra khi mình so sánh biến bằng bằng thì const vẫn có thể áp dụng operator này  
sau khi chạy nó sẽ Input khi mình nhập sai thì sẽ báo not equal. , khi nhập đúng 123456 thì no sẽ equal   
  
Bài 3  
sau khi coi để ta thấy để cho sẽ giống nhứng đề khác có 3 loại class và mỗi loại sạp là 40 tr/^2/ năm thì no sẽ nằm ở class chính, thông tin chung quản lý thì nó cúng nằm trong class chính luôn

và bài này mình cũng sử dụng vector   
trong Class lớn là   
Gọi vector<Sap\*>sap;   
khai báo vòng lặp và nó hỏi mình nhập loại mặt hang   
cin >> temp   
nó sẽ 1 trong 3 case để nhập 3 loại mặt hang   
  
cả 3 đều là hàng nhập   
    
và cái hàm nhập nó ở đây   
cout << “ Hop dong thue sap”  
cout << “Nhap ten chu so huu”  
Thì sau 1 năm nó sẽ hỏi doanh thu  
gọi biến double b để chứa doanh thua 

nhập vô doanh thua và inputDoanhThu   
  
inputDoanhThu nó nằm trong class chính, bỏ vào doanh thua và nó sẽ gán biến dt doanh thu mà mình chuyền vào vào đây  
  
và hàm print 

no sẽ in ra sap thứ mấy, của ông ba nào phải trả bao nhiêu tiền   
  
Catong nó trả về   
return getDonGia()\*getDienTich() + getDoanhThu()\*(this->thuedoanhthu/100)+ getDVDL()(get dịch vụ đông lạnh);  
Mỗi class kế thường cho 1 biến thuế doanh thu  
  
đẻ double luôn mới tính chính xác được tại vì nó chia 100  
cái thue doanh thu của trang thức bằng lấy Doanh Thu để cho   
Nếu muốn chuyên nghiệp hơn thì mình viết thêm check coi số thứ tự đó có ai bán chưa, mà đề ko yêu cầu thì khỏi.