CS 246 S2007 Final

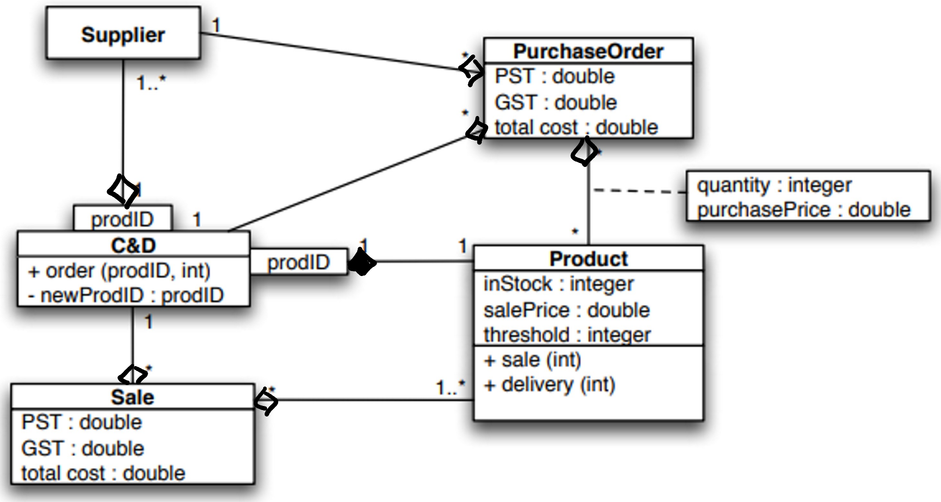
1. a) N/A? b) False c) True d) False e) True f) True g) True h) False i) N/A j) N/A

2. a)

```
void deepCopyVector(const vector<int*> v1, vector<int*> v1Copy) {
        if (v1.size() > v1Copy.size()) {
                for (int i = v1.size() - v1Copy.size(); i >= 0; --i) {
                        v1Copy.push_back(new int);
                }
        }
        while (v1.size() < v1Copy.size()) {
                int lastIndex = v1Copy.size() - 1;
                delete v1Copy.at(lastIndex);
                v1Copy.erase(lastIndex);
        }
        for (int i = 0; i < v1Copy.size(); ++i) {
                *v1Copy.at(i) = *v1.at(i);
        }
}
```

b) a) 1 b) 5 c) 6 d) 8 e) 4 f) 8 g) 7 h) 1 i) 2 j) 3

3. a)



b) An association class makes more sense, since a PurchaseOrder defines attributes of the relationship between the Supplier and C&D?

c) Not covered :P

4. a)

```
class C&D {
        vector<Supplier*> suppliers;
        map<ProdId, Product*> products;
        vector <PurchaseOrder> purchaseOrders;
        vector<Sale> sales;
public:

};
```

b)

```
addProduct(Product* product) {
        ProdId prodId = newProdId();
        products[prodId] = product;
}
```

5.

```
class IntVector {
private:
        std::vector ivec;
public:
        IntVector(int size);
        void setVal(int index, int value) throw(std::out_of_range);
        IntVector* Add (IntVector *v2) throw(std::out_of_range);
};
```

Interface specification not covered :P

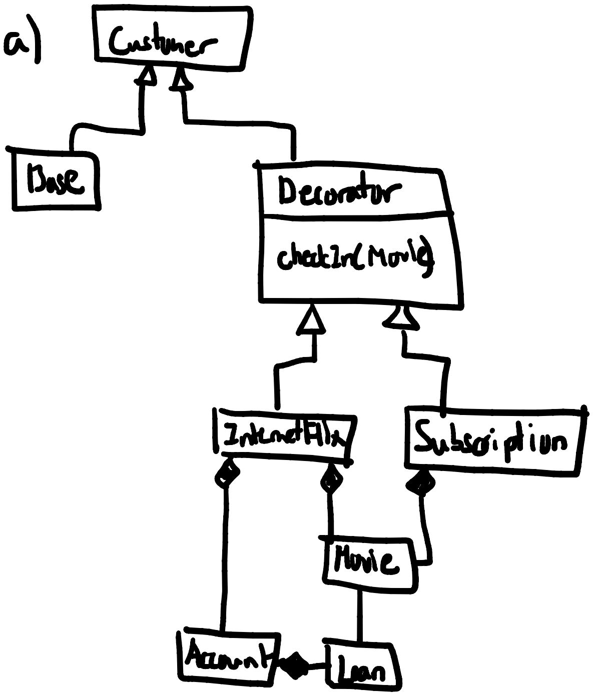
6. a) a) 1 b) 5 c) 4 d) 7

B) a

D) a, c

E) ???

7. a)



b) I am not sure how the Visitor Pattern is relevant here…