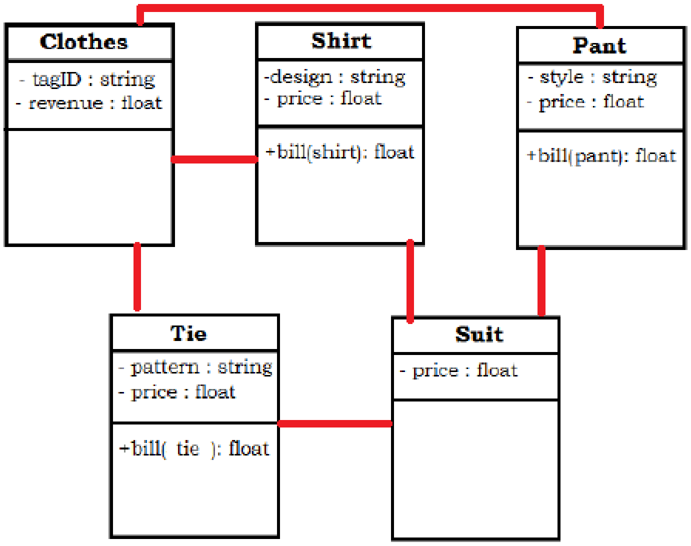
Question 1:

a)



b)

Hierarchical inheritance (Clothes, Shirt, Pant, Tie)

Multi-level inheritance (Clothes, Shirt, Suit) (Clothes, Pant, Suit) (Clothes, Tie, Suit)

Multiple inheritance (Shirt, Pant, Tie, Suit)

c & g)

Clothes:

```
        static float revenue; // initialized to zero outside class
        string tagID;

        Clothes( ) { tagID = 0;}
        Clothes(int tagID) { this->tagID = tagID; }
```

Shirts:

```
        string design;
        float price;
        static int stock; // initialized to some value outside class

        shirt(string design)
        {
                this->design = design;
                price = 1500;  //price is uniform for all styles
        stock--;                        // for Part (g)
        Clothes::revenue += price;      // for Part (g), assuming public access to revenue
    }

Pant:
    string style;
    float price;
    static int stock; // initialized to some value outside class

    Pant(string style)
    {
        this->style = style;
        if(style  == "bellbottomed")
            price = 1200;
        else if(style == "straightfit")
            price = 1600;
        else  if(style == "narrowfit")
            price = 2000;

        stock--;                        // for Part (g)
        Clothes::revenue += price;      // for Part (g), assuming public access to revenue
    }

Tie:
    string pattern;
    float price;
    static int stock; // initialized to some value outside class

    Tie(string pattern)
    {
        this->pattern = pattern;

        if(this->pattern == "stripes")
            price = 700;
        else if(this->pattern  == "checkered")
            price = 800;

        stock--;                        // for Part (g)
        Clothes::revenue += price;      // for Part (g), assuming public access to revenue
    }

Suit:

    float price;

    Suit(string pDesign, string pStyle, string pPattern): Shirt(pDesign), Pant(pStyle), Tie(pPattern)
    {
        price = Shirt::price + Pant::price + Tie::price; // assuming public access
    }
```

d)

```
float bill()
{
    // Shirt s1("formal");
    // Pant p1("straightfit");
    // Tie t1("stripes");

    return Shirt::bill(s1) + Pant::bill(p1) + Tie::bill(t1);
}
```

e)

```
float bill(Shirt& s, string voucher)
{

    float discountRate = 0.4; // assuming a 40% discount

    if(voucher == "XYZ") // assuming it to be a valid voucher
        return s.price*discountRate;
    else
        return s.price;
}
```

f)

```
class Shirt
{
    // rest  of the code
    friend bool operator < (const Shirt&, const Pant&);
};

class Pant
{
    // rest  of the code
    friend bool operator < (const Shirt&, const Pant&);
};

bool operator < (const Shirt& s, const Pant& p)
{
    if(s.price > p.price) // assuming public access
    {
        cout << "This shirt instance gives more profit.";
        return true;
    }
    else
    {
        cout << "This shirt instance does not give more profit.";
        return false;
    }
}
```

h)

```
Pant(const Pant& o)
{
    // assuming public access
    style = o.style;
    price = o.price;

    stock--;
    Clothes::revenue += price;
}
```

i)

```
class Clothes
{
    // other code
    friend void taxCalculation();
};

void taxCalculation()
{
    // can access revenue even when it's private for tax calculation
}
```

Question 2:

```
class A
{
    int x;
    public:
    A(int i)
    {
        x = i;
    }
};
```

```
class B: public A
{
        int y;
        public:
        B(int i, int j): A(i)
        {
                y = j;
        }
};

class C: public B
{
        int z;
        public:
        C(int i, int j, int k): B(i, j)
        {
                z = k;
        }
};

int main()
{
        C ob(2, 5, 12);
}
```