1. Draw a **sequence diagram** for this code.

```
class SaleTest {
    public double testAddItem( ) {
            Item item = new Item("111");
            item.setQuantity( 3 );
            Sale sale = new Sale( );
            sale.addItem( item );
            double total = sale.getTotal();
            return total;
    }
}
class Sale {
    private List<Item> items;
    public double getTotal() {
        // compute total of all items
    }
}
```

2. In a Sequence Diagram, how do we show the initial call to testAddItem( )?

What is this kind of message called? (UML has a name for it.)

3. Draw a Sequence Diagram of what happens when add(m1, m2) is invoked. add(m1, m2) is the *found message* that starts the sequence.

```
class Money {

    double value;
    String currency;
    public Money(double value, String currency) …
    public double getValue() { return value; }

}



class Main {

    static Money add(Money x, Money y) {

        double value = x.getValue() + y.getValue();

        String curr = x.getCurrency();

        return new Money(value, curr);

    }



    public static void main() {

        Money m1 = new Money(50, "Baht");

        Money m2 = new Money(80, "Baht');

        Money m = add(m1,m2);

    }

}
```

4. Show how to write "if" in sequence diagram.

UML 2.0 notation

UML 1.x notation

```
double max(double x, double y){

        if ( x > y ) return x;
        return y;
   }

   // in Purse class
  boolean deposit(Valuable v) {
        boolean full = isFull( );
        if (full) return false;
        boolean result = contents.add( v );
        return result;
}
```

3. We want to have a small Dialog Box that shows how much money is in the Coin Purse.

Call this class PurseBalanceView.

3.1 Draw a UML class diagram showing how to apply the *Observer Pattern* so the PurseBalanceView updates itself whenever the purse balance changes.

3.2 Why don't we just let the Purse directly invoke some method of PurseBalanceView?

4. Draw a sequence diagram of what happens when someone deposits 5 Baht in the purse, showing how the PurseBalanceView is notified and updated.

5. Skytrain Ticket machine (State Machine)

When the Skytrain ticket machine is **idle** (waiting for a customer) it displays "*Please select zone*" on the screen.

A customer presses a Zone button and the ticket machine displays the amount of money to insert, e.g. *30 Baht*. It waits for customer to insert coins.

Each time customer inserts a coin, the display amount decreases, e.g. if you insert 10 Baht coins it displays "*20 Baht"*, then "*10 Baht*", "*0 Baht*".

When the amount decreases to 0, the ticket machine prints a ticket and dispenses the ticket. Then the ticket machine immediately returns to idle state.

At any time the customer can press "Cancel". The machine returns any coins the customer inserted and returns to idle state. If the customer inserts a coin *before* selecting a Zone, the machine immediately returns the coin.

3.1 What are the *states* of the ticket machine?

3.2 What are the *events*?

3.3 What are some *guard conditions*?

3.4 Draw a state chart diagram.