Some of the questions on the first page pertain to the code that starts on the second page.

Questions:

1. What is version control? What do the following commands do?

   ```
   git pull
   git add
   git commit
   git push
   ```

   Is this the correct order in which to use these commands? Are there circumstances when you would use them in a different order?

2. Consider the code that starts on the next page. For each SOLID principle, ask the question:

   Does the code violate this principle?
   If so, on which line(s)?
   How can I fix the code so that it better implements this principle?

3. Consider the code that starts on the next page. Are there any design patterns that can be used to improve or extend it?

4. In the code that starts on the next page, figure out what (if anything) is inherited by each class.
public class Ticket {

    private static int numSales;
    private String event;
    private String buyer;
    private boolean isForSale;

    public Ticket(String event, String buyer) {
        this.event = event;
        this.buyer = buyer;
        isForSale = false;
        numSales++;
    }

    public void returnTicket() {
        buyer = "";
        isForSale = true;
    }

    public void sellTicket(String buyer) {
        this.buyer = buyer;
        isForSale = false;
        numSales++;
    }

    public String toString() {
        return "this ticket for " + event + " belongs to " + buyer;
    }

    public int getNumSales() {
        return numSales;
    }
}

public class TrainTicket extends Ticket {

    private String fromCity;
    private String toCity;

    public TrainTicket(String fromCity, String toCity, String buyer) {
        super("train ride", buyer);
        this.fromCity = fromCity;
        this.toCity = toCity;
    }
}
public void returnTicket() {
    System.out.println("This ticket is for sale again");
}

public String getToCity() {
    return toCity;
}

public void setToCity(String toCity) {
    this.toCity = toCity;
}

public class TwoWayTrip {

    private TrainTicket departTicket;
    private TrainTicket returnTicket;
    private String startDate;
    private String endDate;

    public TwoWayTrip(
        String startDate, String endDate, String fromCity, String toCity,
        String buyer)
    {
        departTicket = new TrainTicket(fromCity, toCity, buyer);
        returnTicket = new TrainTicket(toCity, fromCity, buyer);
        this.startDate = startDate;
        this.endDate = endDate;
    }

    public void printItinerary() {
        System.out.println("Start date: "+ startDate + "; End date: "+ endDate);
    }
}