Practical Object oriented design (Template method and State patterns)

Solve the following problems.

Problem1: Summary of various accounts in bank

The bank you work for (you only work for one bank, of course) offers two kinds of accounts: Savings accounts and Current accounts. The savings account offers an annual interest rate of y (> 0 defined by you) whereas there is no interest offered in the current account. Do the following:

a) A Bank class that is implemented as a Singleton. Therefore, no code in your program should be able to instantiate more than one Bank during a single execution of your code.
b) A bank has multiple Accounts. Create an abstract class called 'Account'. This class will have a Template Method called getAccountSummary( ). This method in turn calls 3 other methods: calcInterest( ), updateBalance( ) and printSummary( ).
c) Create two more classes SavingsAccount and CurrentAccount, that are derived from Account. Both the derived classes implement their own calcInterest( ) method. The methods updateBalance( ) and printSummary( ) are generic to all types of Accounts and are thus to be implemented in the base class 'Account'.
d) Implement any other variables, methods that you feel are needed for these classes to function e.g. you may need a private variable called 'balance'.

Problem2: Connectivity across databases

In the process of writing a database application, suppose you get a new requirement to support both Oracle and SQL Server databases. Both of these systems are based on SQL, the common standard that makes it easier to use databases. We know that in general, when executing queries on these databases, we will use the following steps:

1. Format the CONNECT command.
2. Send the database the CONNECT command.
3. Format the SELECT command.
4. Send the database the SELECT command.
5. Return the selected dataset.

Yet, even though SQL is a common standard at the general level, there are still differences in the details when you will need to use it for supporting the Oracle or the SQL Server database. The specific implementations of the databases differ, however, requiring slightly different formatting procedures for the CONNECT and SELECT commands depending on the type of the database. Do the following:

Practical Object oriented design (Template method and State patterns)

a. Design classes for OracleDatabaseQuery and SQLServerDatabaseQuery which supports method called queryDB.
b. Improve the design by reusing the common functionality of both classes.
c. Improve the design further by reusing queryDB method with the help of template method pattern.

Problem3: Simplified coin-operated washing machine

The washing machine takes $1.00 in quarters to start the machine. When the money has been entered and the start button has been pushed, the machine starts to fill (we will not worry about setting the water temperature for this lab). When the machine is done filling, it begins the wash cycle. After the wash cycle, the water drains and fills again. After filling the second time, the washer starts its spin cycle. When the spin cycle is complete, the washing machine turns off.

The human user may open and close the door when the washer is not engaging it's moving parts (*i.e.* when it is not in the wash or spin cycles). If the door is open when the washer is ready to move on to the wash or spin cycle, an error message is triggered and the washer will not move to the next phase until the door is closed again. In summary, the following actions are generated by the human user:

Insert a Quarter
Push Start Button
Door Open Request
Door Close Request

Problem4: Frequent Flyer Program in Emirates Airlines

Consider a frequent flyer program run by an airline. A traveler starts out at, say, Basic level; then, as miles are accumulated by flying, she moves up to Silver or Gold level. At these levels, travelers can access lounges, get an extra baggage allowance, and also earn miles at a faster rate. For example, at the Silver tier, for every 1000 miles flown, a credit for 1250 miles is earned; at the Gold level, the credit is for 1500 miles. Other benefits also apply, as shown in the sample from Emirates Airlines in the table below:

Services	Emirates Basic	Emirates Silver	Emirates Gold
Access to News Letter	X	X	X
Car Rental Discount	X	X	X
Business Check-In		X	X
Dedicated Call Centre		X	X
Lounge Access			X
Extra Luggage			X
Preferential Seating			X

Practical Object oriented design (Template method and State patterns)

A frequent flyer program works in annual cycles: at the end of each year, a member's activity is calculated for the year and she is assigned to a tier. In the case of our hypothetical program, one of the requirements is to fly 25,000 miles in one year to attain Silver status and keep it for the next year; for Gold status a member must fly 50,000 miles.

a) Design the class for the capturing the complex behavior of FrequentFlyer Object. How do you accommodate the new platinum state introduced by Emirates Airlines in your design?
b) Re-design the above class using state pattern such that new states can be introduced easily.