1. Requirements
A Hot Drinks Machine simulator is required to be analyzed, designed, implemented, tested and documented as outlined below.

Please select a drink

Display

Drink hatch

Coffee
Chocolate
Tea

Coin input

Refund
2. Buying a drink

The following describes the main flow of the simulator:

1. The system displays:

   “Please select a drink”

2. The customer selects a drink by pressing one of the buttons and the system then displays the drink selected and the amount required. Eg. if coffee has been selected the system displays:

   “Coffee selected. Please insert xx Kr”

   Drink prices are different

3. While the customer inserts money, the amount is counted up until the amount inserted is equal to or exceeds the price of the drink.

   While the amount inserted is below the price, the system displays:

   “Please insert yy Kr”

   Where $yy = \text{price} - \text{inserted amount}$

   If the amount inserted is equal to the price of the drink, the system displays:

   “Preparing drink, please wait.”

   If the amount inserted exceeds the price of the drink, the system displays:

   “Preparing drink, please wait. Take xx Kr in refund”

   Assume that the system delegates the actual drink preparation to a subsystem. This system is now waiting for the subsystem to report the drink ready.

4. When the subsystem reports that the drink ready, the system unlocks the drink hatch and displays:

   “Please remove drink”

5. When the system senses that the drink has been removed by the customer the system locks the drink hatch and displays:

   “Please select a drink”

   The system is now ready to service another customer.

Note: The simulator must handle or prevent invalid input.
3. State machine diagram
The report must include the complete state machine diagram and descriptions of all states, events and actions, as exemplified below:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelectDrink</td>
<td>The machine is ready for a new drink to be selected.</td>
</tr>
<tr>
<td>InsertMoney</td>
<td>The machine counts up the amount of money inserted by the customer, and waits for the amount to be equal or greater than the price of the selected drink.</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectCoffee</td>
<td>Coffee has been selected by the customer.</td>
</tr>
<tr>
<td>drinkReady</td>
<td>The brewing subsystem has reported that the drink is ready.</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setDrinkPrice</td>
<td>The price is set according to the selected drink.</td>
</tr>
<tr>
<td>brewDrink</td>
<td>The selected drink is made.</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>

4. Class responsibilities
The report must contain a class diagram and describe the responsibility of each class.

5. Documentation
The report must be written using the report template provided on Campusnet. The Netbeans project should be uploaded onto Campusnet on the specified date. Javadoc must be used to document each class.
6. Implementation

The program should be implemented using the “State” design pattern. The implementation is done in two iterations: Iteration 1 (Assignment 2) and Iteration 2 (Assignment 3).

Iteration 1
The state machine diagram is implemented with a simple console user interface in NetBeans and tested using JUnit.

Iteration 2
The simple console user interface is replaced with a graphical user interface (GUI) using Swing.

7. Test
All tests should be documented in the report and verified by running the tests from the NetBeans project.

8. Delivery
The javadoc documentation for Iteration 1 (Assignment 2) should be uploaded to CampusNet no later than 12.00pm Tuesday 24th November 2009. Iteration 1 should be demonstrated to the supervisor, including JUnit tests on Tuesday 24th November 2009.

Two printed copies of the report should be handed in no later than 12.00pm Friday 18th December 2009. Iteration 2 should be demonstrated on Friday 18th December 2009.

The assignments must be approved in order to obtain admission to the examination.

9. Software
The software should be delivered as a zipped NetBeans project on Campusnet.