COMP201: Software Engineering I
Assignment 1.2 (2018/2019)
(100% mark for Assignment 1.2 is 10% of COMP201 grade)

Deadline for Assignment 1.2: 20th of November 2018, 15:00

OBJECTIVE
This assignment is mainly about "design/implementation".
You should be designing and implementing a simulation of a drinks machine. This simulation will follow the requirements that have been defined in coursework 1 and follow the use cases that you defined. This simulation will not be a full implementation of the original requirement but a cut down basic version with no engineering mode or remote access. All the UI code is provided for you, so there is no GUI programming required. You are strongly encouraged to start this work as soon as possible.

The purpose of this exercise is to understand the challenges of implementing a design from requirements and the state modelling required. This code will be implemented using a simulated version of the hardware, it would then be possible later to implement a real drinks machine be replacing the simulation code with software drivers connected to real hardware.

Assignment number	1 of 2 (part 2)
Weighting	10%
Assignment Circulated date provided to class	24/9/2018
Deadline Day & Date & Time	20th of November 2018 at 15:00 (3PM)
Submission Mode	Electronic submission
Learning outcome assessed	1. Realise the problems in designing and building significant computer systems 2. Understand the need to design systems that fully meet the requirements of the intended users 3. Be able to apply these principles in practice 4. Be able to demonstrate how to effectively implement an O-O design in an O-O language such as Java or Python.
Submission necessary in order to satisfy Module requirements	No
Purpose of assessment Marking criteria	To assess the students ability to analyse, generate and document user requirements See end of document
Late Submission Penalty	Standard UoL Policy

Description of problem

Produce a drinks machine simulation in Java using the existing drinks machine simulation code as a base. (This has been made available to you as an Eclipse project). Also produce state machine charts for coin handling and hot water handling components.

NOTE You only need to fulfil the requirements of this coursework not the whole of the requirement detailed in coursework 1.

The drinks machine needs to do the following:

Coin handling 20%

Accept coin codes and credit the user with the correct credit. If the user presses reject, return the users remaining credit. On return of credit the machine should return the change using the least number of coins possible, for example £1.50 can be returned as a pound coin plus one fifty pence coin. However, if there are no one pound coins in the machine, this may not be possible. Your code will have to handle this logic.

Hot water handling 20%

The machine must control the temperature of the hot water in such a way that it is ready to start making a drink within 4 seconds minimum but switch of the heater as often as possible (to save electricity). You will have to switch the hot water heater on and off depending on the temperature of the water in the heater. If you cannot control the water heater (for example if it increases in heat when you have switched it off), you need to shutdown the machine and display an error message.

SAFETY The machine should not leave the hot water at boiling point 100 C for any length of time, as this is not safe. If the water reaches boiling point the machine should be shutdown.

Drink making 40%

The machine will only make a drink if the correct credit is available and the levels for the relevant drink ingredients are available. The details of the drinks are shown in table 1. If there is not enough credit or there is not enough ingredients the machine should show an appropriate message. When the drink is being made the credit should be reduced by an appropriate amount. If it is impossible to make any of the drinks on the list the machine should not allow coins to be entered and display an error message.

Drink	Code	Cost	Recipe
Black coffee	**101**	**1.20**	**Coffee powder, 2g.** **Hot water 95.9 C**
Black coffee With sugar	**102**	**1.30**	**Coffee powder, 2g.** **Sugar 5g** **Hot water 95.9 C**
White coffee	**201**	**1.20**	**Coffee power, 2g,** **Milk powder 3g** **Hot water 95.9 C**
White coffee with sugar	**202**	**1.30**	**Coffee power, 2g,** **Milk powder 3g** **Sugar 5g** **Hot water 95.9 C**
Hot chocolate	**300**	**1.10**	**Chocolate powder 28g** **Hot water 90 C**

Recipes

Cup sizes
The drinks defined in the list are the small cup sizes, the cups available are small 0.34L , medium 0.45 and large 0.56.
To order a medium cup the user should prefix the code with a 5, for example for medium white coffee the code is 5201, to order a large cup the prefix is 6, so for large white coffee the code is 6201.
The cost of a medium cup is 20 pence more than the small cup drink.
To cost of a large cup is 25 pence more than the small cup drink.
You need to also scale up the amount of coffee, milk, sugar and chocolate to make the larger drinks.

Drink delivery temperature
The actual temperature of the drink delivered should be 80C (+ or -) 4%.
This is to make sure the drink is safe enough to drink but still enjoyable.
The temperatures in table 1 in the recipes section refers to the initial temperature of the water used at the beginning of making the drink, at least 20% of the water should be delivered at this temperature or within 5%.

State charts 20% (see Lecture 7)

You should provide 2 state charts (10% per state chart) for the hot water control component and the coin handling component. These charts should **NOT** have an end state, should have **no deadlocks** (states with no exit criteria) and follow the notation laid out in lecture 7.

Marking Criteria

Part	A++ to A 70%+	B 60-69%	C 50-59	D 40-49	E+ 35-39	E- to G <35
1	All functional tests passed, and code shows excellent design qualitied such as high coherence, encapsulation on and weak coupling	Nearly all functional tests passed and good design structure.	Most functional tests passed but design elements shows some major flaws.	Some critical functional elements missing	A working submission in which but which doesn't pass any of the functional tests	A partly working submission.