**Object Oriented Systems**

**UML Assignments for Laboratory**

**Tool to be used: BO UML**

**You have to draw diagrams for Assignments 1 to 5**

**Submission Deadline: March 28, March 30, 2016**

**Assignment 1.**

A library has a number of subscribers and a catalogue of borrow-able items. There may be a number of copies of each borrowable item. Each copy of a book has a unique accession number. A borrow-able item may be a book or a CD or a micro-film etc. There are two types of subscribers in the system. One would be ordinary and the other one would be privileged. Each Subscriber has a number of tickets. A subscriber can borrow a book for a ticket. Ordinary subscribers have 5 tickets while privileged subscribers have 10 tickets. Each subscriber has a unique number, name, address, the list of items borrowed etc. A catalogue contains a list of borrow-able items. Any borrow-able item has a unique number, and a name. A book also has name(s) of the author(s).

The librarian wants to (i) add a new subscriber, (ii) delete an existing subscriber, (iii) issue a borrow-able item to a subscriber, (iv) accept a borrow-able item returned by a subscriber, (v) add new borrow-able items to the catalogue, (vi) list all subscribers having a copy of a given borrow-able item, (vii) list all borrow-able items which are being currently borrowed by a given subscriber. While issuing a book, the librarian provides a date before which the book must be returned. If the book is not returned by the returned day then an ordinary subscriber is fined @ Re 1/= per day per book. An privileged customer is fined @ Rs 0.50 per day per book.

* Draw a Sequence Diagram for Borrowing Borrowable Items.
* Draw a Sequence Diagram for returning Borrowable Items.
* Define the function “borrow” of subscriber classes in line with the Sequence Diagram. [ Not for laboratory]

**Assignment 2.**

Draw the Statechart Diagram of a *Subject* class during registration. Instances of *Subject* can be in the *Proposed*, *Scheduled*, *Open For Enrollment*, *Full*, and *Closed to Enrollment* states.

Complete the Statechart diagram of the a *Subject* class by considering the states *Being Taught* and *Final Exams.*

**Assignment 3.**

Answer the following questions.

1. A student can take courses at a Univesrity. Courses can be lectures or lab classes. Each student must take at least one course. Each course is held by a research department. Each research department has at least one professor and 0 to 30 research assistants. Students, professors and research assistants have a lot in common: They have a name, a first name and a date of birth. Which diagram type should be used to describe this? Draw such a diagram for this.
2. Model the relationship between a car (that has an engine and a color) and its owners (having a name) in a UML class diagram. A car can have several owners over time, but only one or none owner at a time. Do not forget cardinalities, role names, attributes and their types.
3. A telephone handset remains idle till someone lifts the receiver when the handset becomes active. In active mode, it plays the dial tone. If no digit is dialed for 15 seconds then it plays a message and times out. As long as there is dial tone, a user may dial a number. If the number is invalid then a message is displayed. If the number is valid then the telephone tries to connect. If the network is busy then the handset plays busy tone. Otherwise, the user hears the other side ringing. If the callee answers then the user talks through the handset. From the active state, if the caller hangs up, the handset is disconnected and it becomes idle again. Draw a state chart diagram for the handset.
4. Consider an air conditioner with the states on and off. When it is in OFF state, if the power switch is put on, the air conditioner goes to an ON state. There are two sub-states in ON state: HEATER and COOLER In the HEATER state, if a mode button is pressed, *setCooler* method is called and the new state becomes COOLER after the temperature attains a particular value. In the COOLER state, if a mode button is pressed, *setHeater* method is called and the new state becomes HEATER after the temperature attains a particular value. In the HEATER state the method *heaterOn* is called and in the COOLER state *CoolerOn* method is called. Draw a UML state chart diagram for the air conditioner using **disjoint sub-states**.

**Assignment 4.**

Draw a UML **statechart diagram** that indicates how a watch with a stopwatch feature works. The watch has two buttons on the side of the case: a **mode** button and a **set** button.

* The watch starts out in time-keeping mode, showing the time of day. (The system method showTime() is called to show the time of day.)
* If the **mode** button is pressed once, the watch switches to stopwatch mode (by calling the method showStopwatch() followed by the method clear()). In this mode, if the **set** button is pressed, the stopwatch starts timing (by calling start()). When **set** is pressed again, the stopwatch stops timing (by calling stop()). A third press of the **set** key resets the stopwatch to all zeros (by calling clear()).
* If the **mode** button is pressed while the watch is in stopwatch mode (whether it is currently timing or stopped or cleared), the watch enters time-setting mode, indicated by showing the time with the hours flashing (by calling showTime() followed by flashHours()). The **set** button may then be pushed repeatedly, with each push causing the hours to be incremented (by calling incrHours()). If the **mode** button is pushed again, the watch indicates that it is ready to set the minutes by flashing the minutes indicators (by calling flashMins()); subsequent presses of the **set** button then advance the minutes (with incrMins()). A third press of the **mode** button starts the seconds flashing (flashSecs()); in this mode, a single press of the **set** button resets the seconds to 0 (by calling resetSecs()) and resumes the normal display of the time. If the **mode** button is pressed instead of the **set** button while the seconds indicators are flashing, then the watch resumes the normal display of the time (showTime()) without resetting the seconds.

From this state chart diagram generate the code of the class “Watch”.

**Assignment 5.**

Consider the world of companies: Companies employ employees (who can only work for one company), and consist of one or more departments. Each company has a single president, who is an employee. Departments have employees as members and run projects (one or more.) Employees can work in 1 to 3 projects, while a project can have 2 to 50 assigned employees. You may assume that companies have a name and address, while employees have a emp# and a salary.

Draw a class diagram for the description above. Make sure to show attributes, multiplicities and aggregation associations, where appropriate.

**Assignment 6.**

Read <http://www.uml-diagrams.org/activity-diagrams-examples.html>.

**Assignment 6.**

Consider the following code snippet.

Stack \*st;

Operand\* Expression::evaluate()

{

Token \*t; String s;

while ((s = this-> getNextToken())) != End){

if (s -> getType == OPERAND){

t = new Operand(s);

t->process();

}

else {

t = new PlusOperator(); //PlusOperator is derived from BinaryOperator

t->process();

}

}

return (st->pop());

}

void Operand:: process(){

st->push(this);

}

void BinaryOperator::process(){

Operand \*o1 = st->pop();

Operand \*o2 = st->pop();

st->push(this->eval(o1, o2));

}

void PlusOperator:: eval(Operand \*o1, Operand \*o2){

return (new Operand(o1.getValue() + o2.getValue()));

}

main(){

s1 = new Stack();

Expression \* e1 = new Expression (“56 12 + $”);

e1-> evaluate;

}

Draw the Sequence diagram corresponding to “Expression::evaluate” method