Learning OOP with Robotics[1] :

Bored of listening to lots of theory? Tired of coding imaginary situations? Yes !! Then we would like to try a different learning strategy. Dr. Aegidius Plüss professor[2] for Computer Sciences and their teaching methods at the University of Berne has created a new set of tools for learning programming concepts using innovative solutions . Most of the idea is learning coding by experimentation. Based on the tools developed by Dr. Plüss, Bern University [3] has developed a course work on learning JAVA on Lego Nxt Robot.

We use the same methods to learn major concepts of OOPs in our foundation course.
We start with doing all the tutorials available on the course and running it on a simulation environment. Once we are confident on the simulation environment we move towards the actual hardware i.e. Lego Robot. Here we will understand the difficulties in the moving from simulation to actual hardware. The final aim will be to create an autonomous robot which can do assigned tasks successfully.

Hope we will be able to learn a lot and be successful in creating an autonomous robot :) .

Submission :

Simulation Introduction :

Submission Deadline : Monday evening

Topics:

1. **Class diagram of Robot Sim class** . The doc of robot sim is available here http://www.aplu.ch/classdoc/robotsim/index.html . Please learn what an class diagram is . For a sample of what a class diagram is Please scroll down on http://www.aplu.ch/classdoc/robotsim/index.html and detailed class diagram of same http://www.aplu.ch/home/nxtjlib/klassendiagramm.html
2. A small write up on :
   what is a class and what is an interface (Explain with the RobotSim package)?
   Specify when will you make a class and when will you use an Interface ?
3. Implementation of all the tutorials available in RobotSim
   http://www.java-online.ch/lego/legoEnglish/index.php?inhalt_links=nxtSim/nav_nxtSim.inc.php&inhalt_mitte=NxtSim/move.inc.php
   A Github repository will be created and all the tutorial code will be pushed in repo.
   Coding standard (Explained in class) should be maintained.
   Explore Javadoc ?

All the explanation of the code should be added in the code as comments . (Try not to copy the code, but to read the code and then code by hand, also the comments.)

4. Submission of the Exercises. Same procedure with Github.
   Each exercise folder submission will contain the following elements:
   a. src folder with you source file
   b. doc folder with a diagram/note explaining the functionalities
   c. test (if you have some output like screenshot)

Simulation Advanced:

Submission Deadline : Wednesday
Topics : Assignments provided

On Hardware - Legos :

Submission Deadline :
Topics : Running simulation topics on hardware

Hackathon - Tasks :

Submission Deadline : Friday - 25th September
Daily status meeting : timing TBD
Competition : Friday 25th September

- Latex Presentation
- Design
- Git workflow
- code
- Running example
- 20 min

Topics :

1. Exploring out of box :

Level : Easy
Task :

The robot will be placed inside a closed box marked by thick black line on ground with one outlet. The robot has to search for the outlet and try to exit the box as early as possible.

Things to learn :

- Use of sensors
- Good navigation planner

Competition : All the teams will be timed on their timings to complete the tasks.

2. Line Follower :

Level : Medium
Task :

The robot has to follow a black line with complex turns. Robots will be timed on different tracks.

Things to learn :

- Use of sensors
- Control design (P, PI, PID)

Competition : All the teams will be timed on their timings to complete the tasks.

3. Navigate in a rectangle

Level : Hard
Task :

The robot has to move in a rectangular path of 100 cm x 50 cm.

The robot has to complete 3 rounds of the rectangle. It has to do movement only using internal sensors.

Things to learn :

- Use of multiple sensors
- Kalman filter
- http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/
- http://www.lejos.org/nxt/pc/api/lejos/util/KalmanFilter.html
- https://nxttime.wordpress.com/2010/03/15/cracking-the-kalman-filter/

Competition : All the teams will be timed on their timings to complete the tasks.

Appendix

download Links :
JGameGrid: http://www.aplu.ch/home/download/JGameGrid.zip
NxtJLib: http://www.aplu.ch/home/download/NxtJLibA.zip

Site :
NxtJLib : http://www.aplu.ch/home/apluhomex.jsp?site=27
JGameGrid : http://www.aplu.ch/home/apluhomex.jsp?site=45

Documentation :
NxtJLib : http://www.aplu.ch/classdoc/nxtjliba/index.html
JGameGrid (only RobotSim) :
http://www.java-online.ch/lego/legoEnglish/index.php?inhalt_links=nxtSim/nav_nxtSim.inc.php&inhalt_mitte=NxtSim/iframeDocNxtSim.html

ToDo for Lab :

You need to complete the sample code available in the section of robotsim in the site below
Bern University of Teacher Education :
http://www.java-online.ch/lego/legoEnglish/index.php?inhalt_links=nxtSim/nav_nxtSim.inc.php&inhalt_mitte=NxtSim/nxtSim.inc.php

References :

1. http://www.aplu.ch/classdoc/robotsim/index.html
2. http://www.aplu.ch/home/apluhomex.jsp
3. http://www.java-online.ch/lego/legoEnglish/index.php?inhalt_links=home/nav_home.inc.php&inhalt_mitte=home/home.inc.php