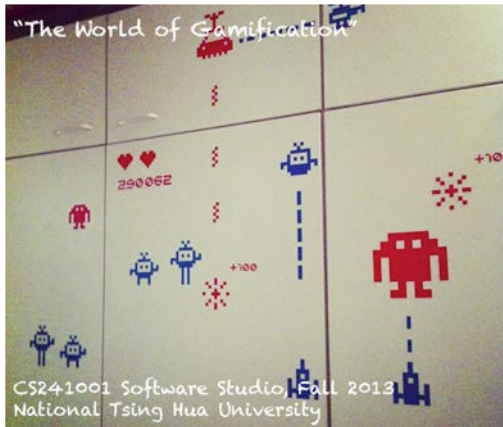| 課程資訊				
科號 Course Number	CS241001	學分 Credit	2	人數限制 Size of Limit	90
開課年級 Course Level	資工系大二			先修科目 Prerequisite	Basic computer programming skills
中文名稱 Course Title	軟體實驗				
英文名稱 Course English Title	Software Studio				
任課教師 Instructor	王浩全				
授課語言 Language					
上課時間 Time	**Tue 3:30pm - (Lab) /** **Thu 3:30pm - (Lecture)**		**上課教室** **Room**	資電 326 (Lab) 台達 109 (Lecture)	

Image source:
http://farm9.staticflickr.com/8464/8105567951_a3672dc01b_z.jpg

Software Studio

NTHU CS Undergraduate Course
Fall 2013
Instructor: Hao-Chuan Wang 王浩全
Email: haochuan@cs.nthu.edu.tw
Course site: NTHU LMS (http://lms.nthu.edu.tw)
TA: 賴建同, 姚道亨, 姜翰廷, 施博瀚, 周佩璇

**Note: This syllabus is tentative. The content of this co
subject to change.**

1. Course Description

The course presents hand-on labs for the software aspect for students to be familiar with software development processes and techniques. The course aims to use sample topics and issues in the area of Human-Computer Interaction (HCI) as a software development 'sandbox' for educational purposes. The main goal of this class is to endow students with experience and skills in software development. Students will still learn some ideas in HCI design in the second half of the course, while this course is not intended to be a comprehensive, systematic introduction to HCI.

As a tentative plan, the class is divided into three parts, including 'fundamentals', 'visualization and interaction' and 'gamification design'. The fundamentals introduce the Java programming language. We will highlight the concept of Object-Oriented Programming (OOP) and help students understand the costs, benefits and value of this programming paradigm. We will introduce useful productivity tools for basic software development, such as the Eclipse IDE. The

second and third parts of this class look at using Java-based tools (e.g., Processing) to design, prototype and experiment with design ideas of interactive and social computing in HCI. The final term project will be a larger software project that requires 4-5 students to turn any regular tools or systems into computer games. This sort of "gamification" design approach aims to increase user's experience and enjoyment to interact with computers. Also, gamification can be a useful method to trigger people's interest in making contributions and complete work that's difficult or expensive to do. Students project groups will actively brainstorm and identify their gamification targets, and creatively design and implement the game-based systems that they propose.

Basic understanding of computer programming (using C or other programming languages) and data structure is required.

2. Textbook
Lecture notes

3. References
Java:
- Head First Java, 2nd ed. O'reilly / 深入淺出 Java 程式設計. 碁峰.
- Oracle's Java tutorial http://docs.oracle.com/javase/tutorial/
- Tutorials for Java beginners
  http://heather.cs.ucdavis.edu/~matloff/java.html
  http://www.cs.utexas.edu/~lavender/courses/tutorial/index.html

Eclipse and SVN:
- Eclipse and Java for Total Beginners
  http://eclipsetutorial.sourceforge.net/totalbeginner.html
- How to use Subversion with Eclipse
  http://www.ibm.com/developerworks/opensource/library/os-ecl-subversion/

Visualization and interaction programming:
- Processing – Java-based language for visual programming http://processing.org/
- Tutorials for processing
  http://www.learningprocessing.com/tutorials/

4. Teaching Method
Lecture and Lab

5. Tentative Content
PART I- Fundamentals
- Introduction and logistics: Overview of course goals, structures and requirements. Overview of tools and development environments
- Java basic- syntax, control statements, method, array, basic object-oriented programming
- Java advanced- event handling, exception, thread, I/O, network
- Team collaboration and version control

PART II- Visualization and Interaction
- *Processing*: Programming language for visualization and interaction design
- Visual communication of information

- Simple interaction design and prototyping

PART III- Gamification Design
- Problem analysis: Basic ideas of user and task analysis and interaction design
- Prototyping gamification: What make games interesting and engaging?
- Human computation and crowdsourcing: Turn works into games and vice versa.

6. Weekly Schedule

Week	Lab (3:30pm Location: 資電 326)	Lecture (3:30pm Location: 台達 109)
1	*Sep 17* Overview of course logistics, requirement and policy. Software development and HCI	*Sep 19* No Class 中秋節
2	*Sep 24* Lab Overview: Introduction to JDK and IDE	*Sep 26* Java 1: Variable, Conditional Statement, Loop
3	*Oct 1* Java Lab 1: Java Procedural Programming	*Oct 3* Java 2: Basic OO Class, Object, Inheritance
4	*Oct 8* Java Lab 2: OOP	*Oct 10* No Class / Professor Attending 兩岸清華大學研討會
5	*Oct 15* Java Lab 3: OOP II User Interface	*Oct 17* Java 3: OO II Interface, Reference, Event Handling
6	*Oct 22* Java Review I	*Oct 24* Java 4: Exception, Specific Java APIs
7	*Oct 29* Java Lab 4	*Oct 31* Java 5: Specific APIs, Thread etc.
8	*Nov 5* Java Lab 5	*Nov 7* Java 6: Coding practices
9	*Nov 12* Java Review II	*Nov 14* Midterm Exam 台達 109
10	*Nov 19* Processing Lab 1	*Nov 21* InfoVis: Intro to Processing, Visualizing Data and Interaction
11	*Nov 26* Processing Lab 2: Integrating Java and Processing	*Nov 28* Project Management: Techniques and Tools- GitHub / Sample Proposal 繳交專案分組
12	*Dec 3* Sample final project demo (by TA) / Project Management Lab 1	*Dec 5* Project Brainstorming Professor Attending ACM CHI PC Meeting at Toronto

13	*Dec 10* Question answering	*Dec 12* Gamification: Overview of Interaction Design, User Experience and Gaming 繳交專案計劃書(project proposal)
14	*Dec 17* Project Time	*Dec 19* User-Centered Evaluation
15	*Dec 24* Anonymous Lab	*Dec 26* Extra Topic 繳交進度報告(progress report)
16	*Dec 31* Project Time	*Jan 2* Project Time
17	*Jan 7* Project System Demo	*Jan 9* Project System Demo Jan 14 繳交結案報告(final report)

7. Evaluation
Midterm Exam 15%
Lab Assignments 48% (8 assignments)
Term Project 27% (Proposal 5%, System Demo 10%, Final Report 12%)
Participation 10% (Course attendance and in-class submissions)

8. Honor Code
Any cheating will be handled seriously in compliance with the university rules. All assigned work is expected to be individual, except where explicitly written otherwise (e.g., term project). You are encouraged to discuss with your classmates; however, what you hand in should be your own work.