Course Syllabus

Name of Institution: Walailak University
Campus/Faculty/Division: International College
Digital Innovation Engineering (International Program)

General Information

1. **Course**
   - Code: **DIN61-222**
   - Name: **Advanced Programming**

2. **Credits**
   - 3 (2-2-5)

3. **Programme**
   - Bachelor of Engineering; Program in Digital Innovation Engineering (International Program)

4. **Coordinators and lecturers**
   - Andrew Davison
   - Dept. of Computer Engineering, Prince of Songkla University (PSU),
     Hat Yai, Songkhla 90112
   - `ad@fivedots.coe.psu.ac.th`
   - `http://fivedots.coe.psu.ac.th/~ad/`

5. **Semester/Year of Study**
   - 1/2019

6. **Prerequisites**
   - DIN61-114 "Data structures and algorithms"
   - Students should have prior knowledge on basic mathematics, hardware, and software.

7. **Co-requisites**
   - None

8. **Times/Places**

-	Lecture:	Day: Saturday	Time: afternoon (2 hours)	Room: ??
-	Lecture:	Day: Saturday	Time: evening (2 hours)	Room: ??
-	Lecture:	Day: Sunday	Time: morning (1-2 hours)	Room: ??
-	Lab:	Day: Sunday	Time: afternoon	Room: ??

Note: these times may vary depending on the amount of course material that needs to be covered, and on student feedback.

9. **Last Update**
   - July 12, 2019 (version 2)

Aims and Objectives

1. Aims

At the end of the semester, students should:

1.1 Have developed advanced programming skills related to OOP;
1.2 Have an understanding of object oriented modeling;
1.3 Be able to program in Java to an advanced level.

2. Objectives

- knowledge, understanding, analysis, practicality, synthesis and integration, responsibility, creativity, self-actualization.

Description and Implementation

1. Course Description

- This course extends the study of basic programming principles introduced in DIN61-111 "Programming Fundamentals". Advanced concepts of program design, implementation and testing will be introduced within a framework of object oriented programming using Java.

2. Teaching Hours

Lectures	Labs	Self-study
30	30	75

3. Academic Advising Time for Students

- During classes and labs.
- Any time by e-mail.

Improving Student Learning Outcomes

1. Morals and Ethics

1.1 Morals and ethics to be developed:

- Help students understand the learning task, and to make connections to past and present learning experiences.
- Stimulate interest and prompt students to identify their own questions about the subject.
- Enable them to use small group discussions to stimulate and share ideas.
- Help students to connect previous knowledge to the new concepts introduced in the subject.

1.2 Teaching methods

- Lectures, exercises, a programming project, and labs.

1.3 Evaluation methods

- Frequency of attendance and submission punctuality.

2. Knowledge

2.1 Knowledge to be acquired:
- OOP: Classes and Objects; Inheritance, polymorphism; Abstraction
- Object Modelling: Object Interaction; Grouping Objects; Good Class Design
- Java Concepts: Imperative features; Using Libraries; GUI; Exception handling

2.2 Teaching methods
- Lectures, exercises, a programming project, labs.

2.3 Evaluation methods
- Two exercises, a programming project, two exams.

3. Cognitive Skills

3.1 Cognitive skills to be developed:
- To enhance a student's ability to solve problems, think critically, and reason scientifically.
- To raise the student's confidence levels in their problem-solving abilities.

3.2 Teaching methods
- Lecturers, exercises, a programming project, labs.

3.3 Evaluation methods
- Measurement by problem solving.

4. Interpersonal Relationship Skills and Responsibilities

4.1 Interpersonal relationship skills to be developed:
- To be a good listener.
- Working creatively with others, communicating clearly.
- Submit exercises, the project, and other required work, on time.

4.2 Teaching methods
- Lecturers, exercises, a programming project, labs.

4.3 Evaluation methods
- Measurement by problem solving.

5. Numeric analytical skills, communication skills, and IT skills

5.1 Skills to be developed:
- Mathematical and scientific skills.
- Computer and internet skills.

5.2 Teaching methods

- Lecturers, exercises, a programming project, labs.

5.3 Evaluation methods
- Measurement by problem solving.

Teaching Plans

1. Teaching Plan (part 1)

Week	Topic*	Teaching^ hours	Labs^ hours	Self-study ;hours	Evaluation #
1	Preliminaries; Java Introduction	5	1	4	
2	Classes and Objects	--	2	4	
3	Object Interaction	--	2	4	
4	Grouping Objects	5	2	6	
5	Using Libraries	--	3	6	Ex 1 begins
6	Good Class Design	5	3	6	
7	Inheritance	--	1	9	Ex 1 due in (10%)
8					**Midterm Exam (25%); 2 hours**
9	Inheritance	--	1	4	
10	Inheritance	5	1	3	
11	Abstraction	--	2	5	Ex 2 begins
12	GUIs	5	3	4	
13	GUIs	--	3	4	Ex 2 due in (10%); Project begins
14	Layout	5	5	6	
15	Exception Handling	--	1	10	Project due in (20%)
16					**Final Exam (35%); 3 hours**
	Totals:	30	30	75	100%

* All topics will be taught by Andrew Davison.

^ All teaching will use a combination of lecturing and in-class discussions. The labs will examine the code used in the slides, in the j9fp textbook (see "Teaching Materials" below), and give time for the students to wotk on the programming parts of the exercises and the project.

For other evalution criteria, see the "Evaluation and Improvement of Course Management" section below.

Teaching Plan (part 2)

Week	Learning Objectives	21C Skills*	Materials/ Resources^	Readings#
1	Java Introduction	3-10	Slides: 01. prelims and 02. simple	j9fp: ch 2
2	Classes and Objects	3-10	Slides: 03. classes	j9fp: ch 3
3	Object Interaction	3-10	Slides: 04. interactions	j9fp: chs 4 & 5
4	Grouping Objects	3-10	Slides: 05. grouping	j9fp: ch 7
5	Using Libraries	3-10	Slides: 06. libraries	j9fp: ch 14
6	Good Class Design	3-10	Slides: 07. design	Slides: 16. style
7	Inheritance	3-10	Slides: 08. inherit-1	
8				
9	Inheritance	3-10	Slides: 08. inherit-1 and 09. inherit-2	j9fp: ch 9
10	Inheritance	3-10	Slides: 09. inherit-2	
11	Abstraction	3-10	Slides: 10 abstract	j9fp: ch 10
12	GUIs	3-10	Slides: 11 guiIntro and 12 gui-1	ch12 from previous edition
13	GUIs	3-10	Slides: 13. gui-2	
14	Layout	3-10	Slides: 14. layout	
15	Exception Handling	3-10	Slides: 15. excpt	j9fp: ch 11
16				

* 21st Century skills are numbered according to the legend: 1. Literacy; 2. Numeracy; 3. Reasoning; 4. Problem Solving; 5. Critical Thinking; 6. Collaboration; 7. Communication; 8. Computing; 9. Career and Life skills; 10. Cross-Cultural Skills

^ All the materials and resources for this course are itemized in the "Teaching Materials" section of this document.

j9fp refers to the textbook "Java 9 for Programmers" cited in the Teaching Materials section.

Teaching Materials

1. **Materials: software, hardware**
   - Liberica JDK 12.0.1 programming language (available at `https://bell-sw.com/pages/java-12.0.1/`)
   - Simple text editor e.g. Notepad++ (available at `https://notepad-plus-plus.org/download/v7.7.1.html`)
   - Whiteboards with pens and eraser
   - Computer with installed Java (see above), Notepad++, and PowerPoint

- Projector, connected to the computer
- Chair, for sitting on by the tired Ajarn.

2. Essential Resources

- Lectures notes in the form of PowerPoint slides are available online at `http://fivedots.coe.psu.ac.th/~ad/teaching/AP222/`
  The code described in the slides is also online at the course website.

- Course textbook: "Java 9 for Programmers", Paul J. Deitel and Harvey Deitel, Prentice Hall, 4th edition, 2017 (**j9fp** for short). Code examples are available at: `http://www.deitel.com/Books/Java/Java9forProgrammers/tabid/3686/Default.aspx`.
  Note: the teacher has a copy of this book, which he will make available to the students.

- The current edition of j9fp does not explain the Swing GUI library. For that part of the course, I will make chapter 12 of the previous edition of the book available to students.

3. Recommended Resources

- Internet (Online books, lectures, tutorials).

Evaluation and Improvement of Course Management

1. Course Evaluation by Students

- Students will evaluate the course effectiveness and the lecturer using an online form provided by the university.

2. Strategies for Teaching Evaluation

- Exercises, a programming project, and exams.

3. Improvement of Teaching

- To inspire.

4. Verification of Student Achievement in Class

- Discuss students' performance with the Dean of WUIC, the Associate Dean of WUIC, and those responsible for the engineering program.

5. Course Review and Improvement Plan for Course Effectiveness

- Discuss teaching methods, activities, evaluation, and students' performance with the Dean of WUIC, the Associate Dean of WUIC, and those responsible for the engineering program.