

1. Course Title (Course Code)

Computer Programming Exercises (2209)

2. Instructor

Markon Sandor

3. Term

Fall 2

4. Outline and Objectives

Programming is the foundation of every other subject in ICT. By becoming proficient in programming, students will be able to actively participate in projects involving system creation. Programming is also necessary for testing ideas, constructing and maintaining networks and servers, and in many other areas.

The course first reviews the fundamentals of procedural programming through experimental exploration, using the dynamic, interactive Python language. Next, students are introduced to static typing using C in the Linux environment and Java in the Processing framework, so they will be prepared to continue their studies with Java for Android programming, and C for generic programming.

The course also introduces students to other common languages like C++ for Linux programming, Common Lisp for AI (Artificial Intelligence) programming, Mathematica for scientific programming etc. so they can continue extending their repertoire after finishing the course.

5. Goals (Attainment Targets)

- (1) Become able to read, understand, and modify programs written in Python and Java.
- (2) Become able to use the development environment of Processing.
- (3) Become able to develop a small but complete system in Processing.
- (4) Can write and use Python scripts for everyday tasks.

6. Correspondence relationship between Educational goals and Course goals

Educational goals		Course goals	
High level ICT skill	Basic academic skills	(1), (2), (3), (4)	
	Specialized knowledge and literacy	(2), (3)	
Human skill (Tankyu skill)	Ability to continually improve own strengths	(3), (4)	
	Ability to discover and resolve the problem in society	Problem setting	
		Hypothesis planning	
		Hypothesis testing	
		Practice	
	Fundamental Competencies for Working Persons	Ability to step forward	(4)
		Ability to think through	(4)
Ability to work in a team			
Professional ethics			

7. Requirements

Foundations of Computer Systems (2206)

8. Textbooks

None. This course uses only material freely available on the Internet (links will be provided on Moodle).

9. Reference Books

None

10. Evaluation

Goals	Evaluation method & point					
	term-end exam	quiz	report	presentation	deliverable	other
(1)			○			
(2)			○			
(3)				○	○	
(4)				○	○	
Allocation			40	30	30	

Course Schedule

(Notice) This schedule is a tentative plan, there might be changes, additions, revisions etc. at the time of delivering the course.

Lesson 1: Orientation, introduction, motivation (lecture and demonstration, 90 min)

What is programming? We consider solving everyday tasks, first by natural language, then by writing exact and detailed instructions. Students learn the basics of the Python language through a few simple exercises and use it to test their understanding of the elements of programming principles.

Lesson 2: Analyzing an Open Source program (group work, 90 min)

A small real-world C program is reviewed in detail, motivating the need for “Code Reading” skills and tools. We discuss the differences of the C syntax from Python, the need for declarations and compilation, and other C-related issues.

Lesson 3: Code Reading fundamentals, review of the program (lecture, 90 min)

We review some commonly used tools and techniques for code reading, and use them to analyze a larger piece of OSS system.

Lesson 4: C programming topics and pitfalls (group work, 90 min)

We work through a set of examples from OSS systems showing how C is used in practice, with emphasis on avoiding bad coding, dangerous constructs.

Lesson 5: Java programming topics 1. (lecture and group discussion, 90 min)

Java is introduced as an object-oriented and safer alternative to C. We review the object-oriented way of thinking and its use to develop models for systems.

Lesson 6: Java programming topics 2. (lecture and group discussion, 90 min)

We review some important Java libraries used in Android programming, including graphics, user interactions, and networking.

Lesson 7: Java programming topics 3. (lecture and group discussion, 90 min)

We review the Processing language and programming environment, in particular its use for rapid prototyping of interactive programs. Students develop original demonstrations with graphics and sound, using Processing.

Lesson 8: Java exercises 1. (group work, 90 min)

Groups of students develop Java applications for Android, using Eclipse.

Lesson 9: Java exercises 2. (group work, 90 min)

Presentation of the group work results.

Lesson 10: Python topics 1. (lecture and discussion, 90 min)

The power of the Python language is demonstrated through using it for common tasks. Reviewed concepts include dynamic development, advanced data types, iterations, exceptions, objects, modules.

Lesson 11: Python topics 2. (lecture and discussion, 90 min)

We review Python libraries, including networking, graphics, numerical computation, databases.

Lesson 12: Python topics 3. (lecture and discussion, 90 min)

We review practical development in Python, with structuring and building a project, testing, using the unit test facilities, and deployment.

Lesson 13: Python exercises 1. (group work, 90 min)

Student groups use Python to build and test a small but complete system providing a well-defined new functionality, in a given short time.

Lesson 14: Python exercises 2. (group work, 90 min)

Presentation and discussion of the results of the group work.

Lesson 15: Wrap-up (discussion, 90 min)

Students review their experience with programming; discussions are directed to helping to make programming a natural, integral part of their life with ICT. We also discuss other programming languages and methods including Mathematica and Common Lisp, and the choices available for finding the best tools.