

AUDIO APPLICATION PROGRAMMING

COURSE OUTLINE

1. GENERAL			
SCHOOL	School of Music and Optoacoustic Technologies		
ACADEMIC UNIT	Department of Music Technology and Acoustics		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0807.5.006.1	SEMESTER	1st
COURSE TITLE	Audio Application Programming		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	WEEKLY TEACHING HOURS	CREDITS
	0	7
Total	0	7

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/SMOT160/

2. LEARNING OUTCOMES
Learning outcomes
The aim of the course is to provide a solid foundation in digital signal processing and the development of audio applications, with an emphasis on applications that require real-time processing. Upon successful completion of the course, students will be able to: - Understand the fundamental components involved in the development of audio applications. - Understand the programming practices required to access and configure audio hardware (e.g., number of channels, block size, sampling rate, etc.). - Be familiar with the programming tools and platforms used for the acquisition, analysis, synthesis, playback, and processing of digital audio signals. - Adapt digital signal processing methods to audio applications that require real-time processing. - Develop audio applications for desktop and mobile computing platforms.
General Competences
The course aims to develop the following general competencies: - Ability to search for, analyze, and synthesize data and information using appropriate technologies. - Ability to adapt to new situations. - Ability to work independently. - Ability to work effectively as a member of a team. - Ability to promote free, creative, and inductive thinking.

3. SYLLABUS

Theory**Python**

Variables

Lists, dictionaries, sets, and tuples

Functions and classes

Memory management

Audio files and digital audio encoding

Real-time audio

C++

Variables, functions, and classes

Pointers and vectors

Vector processing using functions

The JUCE framework

Programming audio effects in C++

Developing real-time audio applications with JUCE

Laboratory Exercises**Python**

Variables and dictionaries

Programming static sound generators

Functions, classes, and sound generators

Memory management and audio encoding

Implementing simple effects for real-time audio processing

C++

Variables, functions, and classes

Vector operations using pointers

Implementing a distortion effect

Implementing sound generators with real-time parameter control

Implementing a delay effect

Implementing a tremolo effect

Implementing a chorus effect

4. TEACHING and LEARNING METHODS - EVALUATION**DELIVERY**

Face-to-face, Distance learning, etc.

On-site

USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Use of ICT in teaching, laboratory education, communication with students

Google colab, eclass, JUCE, Microsoft Visual Studio Community Edition

TEACHING METHODS

The manner and methods of teaching are described in detail.

Activity**Semester workload****Course total****STUDENT PERFORMANCE EVALUATION**

Description of the evaluation procedure

Intermediate and final exercises.

5. ATTACHED BIBLIOGRAPHY

