



## COURSE DESCRIPTION CARD - SYLLABUS

Course name

GUI Programming [S1MNT1>D-PAO]

### Course

Field of study

Mathematics of Modern Technologies

Year/Semester

2/4

Area of study (specialization)

–

Profile of study

general academic

Level of study

first-cycle

Course offered in

Polish

Form of study

full-time

Requirements

elective

### Number of hours

Lecture

15

Laboratory classes

30

Other

0

Tutorials

0

Projects/seminars

15

### Number of credit points

4,00

### Coordinators

dr Grzegorz Oleksik

grzegorz.oleksik@put.poznan.pl

### Lecturers

### Prerequisites

Knowledge and skills from the courses Information Technology I and II, Introduction to programming and Programming with elements of machine learning. The ability to effectively self-educate. Knowing the limitations of one's own knowledge and understanding the need for further education.

### Course objective

The course aims at mastering the basic knowledge and skills in the field of programming GUI by the student using the Python language as an example.

### Course-related learning outcomes

Knowledge:

- knows and understands at least one programming language or programming environment or software package [K\_W07(P6S\_WG)];
- knows and understands the relationship between mathematics and modern technologies [K\_W05(P6S\_WG)];
- knows and understands selected areas of mathematics to an advanced degree and has detailed knowledge of the applications of mathematical methods and tools in engineering and technical sciences [K\_W09(P6S\_WG)].

#### Skills:

- is able to construct an algorithm for solving a simple engineering task and to implement and test it in a selected programming environment [K\_U04(P6S\_UW)];
- can apply modern technologies to solve mathematical and engineering-technical problems [K\_U05(P6S\_UW)];
- can use mathematical tools to support and develop modern technologies used in engineering and technical sciences [K\_U06(P6S\_UW)];
- is able to operate devices, tools, etc. in accordance with general requirements and technical documentation; knows how to apply the rules of occupational health and safety [K\_U11(P6S\_UW)];
- can work individually and in a team and interact with other people; is able to estimate the time needed to complete the assigned task; is able to develop and implement a work schedule that ensures meeting the deadline. [K\_U17(P6S\_UU)];
- can independently plan and implement self-education in order to improve and update their competences [K\_U17(P6S\_UU)].

#### Social competences:

- is ready to critically assess the level of his/her knowledge in relation to research in exact and natural sciences as well as engineering and technical sciences [K\_K01(P6S\_KK)];
- is ready to deepen and expand knowledge to solve emerging technical problems [K\_K02(P6S\_KK)].

### Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures: knowledge is verified on a written test;

Laboratory classes: knowledge and skills are verified on the basis of the evaluation of the programming colloquium;

Projects/seminars: knowledge and skills are verified based on the assessment of project implementation, its functionality and goals;

Grading scale: 0%-49% - NDST, 50%-59% - DST, 60%-69% - DST+, 70%-79% - DB, 80%-89% - DB+, 90%-100% - BD.

### Programme content

introduction to object-oriented programming, regardless of the specific language programming

- data encapsulation
- Inheritance
- polymorphism
- special methods
- decorators
- selected graphical user interfaces
- design patterns

### Course topics

#### Lectures:

- introduction to object-oriented programming (regardless of the specific programming language);
- introduction to object-oriented programming in Python;
- data protection: encapsulation and access modifiers;
- introduction to UML diagrams;
- working with an object: creation, initialization, deletion;
- the issue of inheritance and the concept of polymorphism;
- classes and abstract methods;
- basics of GUI development: tkinter;
- selected GUI development packages: PyGUI, PyQt5,
- exception Handling;
- design patterns (MVC, Strategy)

#### Lab:

- creating your own classes with methods - basics
- creating parents and child classes
- creating classes with special methods
- using decorators in class methods
- creating your own GUI using libraries (tkinter, PyGui)
- using design patterns (MVC, Strategy)

## Teaching methods

Lectures: multimedia presentation, presentation illustrated with examples given on the board, problem solving, multimedia show, demonstration;

Laboratory classes: solving practical problems, discussion, individual or team work;

Projects/seminars: multimedia presentation, discussion, individual or team work, solving practical problems. On the eKursy platform, studies on classes are posted, which can be downloaded by the student.

## Bibliography

Basic:

- Lutz, M., Python. Wprowadzenie, Helion, najnowsze wydanie;
- Steven F. Lott, Dusty Phillips: Programowanie zorientowane obiektowo w Pythonie. Helion 2023;
- Irv Kalb: Python zorientowany obiektowo. Programowanie gier i graficznych interfejsów użytkownika, Helion 2022.

Additional:

- Nathan Metzler: Python GUI Programming with PyQt: A Beginner's Guide to Python 3 and GUI Application Development, Independently Published 2019;
- Eric, M., Python. Instrukcje dla Programisty, Helion, najnowsze wydanie;
- [http://wazniak.mimuw.edu.pl/index.php?title=Programowanie\\_obiektowe](http://wazniak.mimuw.edu.pl/index.php?title=Programowanie_obiektowe);
- <https://analitik.edu.pl/programowanie-objektowe-w-python/>;
- [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)

## Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50