KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Course title

Programming

ECTS code

13.2.0421

Name of unit administrating study

Faculty of Mathematics, Physics and Informatics

Studies

faculty	field of study		
Faculty of Mathematics, Physics and Informatics	Quantum Information Technology	type	all
		form	all
		specialty	all
		specialization	all

Teaching staff

dr hab. Piotr Gnaciński; dr Piotr Mironowicz

Forms of classes, the realization and number of hours

Forms of classes

Laboratory classes

The realization of activities

classroom instruction, online classes

Number of hours

Laboratory classes: 30 hours

ECTS credits

3

3 ECTS

The academic cycle

2022/2023 winter semester

Type of course

obligatory

Language of instruction

english

Teaching methods

- problem-focused lecture
- project-based method (research, implementation, practical project)

Form and method of assessment and basic criteria for eveluation or examination requirements

Final evaluation

Graded credit

Assessment methods

- (mid-term / end-term) test
- assignment work – project or presentation

The basic criteria for evaluation

Project grade: 60%
Test grade: 40%

Method of verifying required learning outcomes

Required courses and introductory requirements

A. Formal requirements

none

B. Prerequisites

none

Aims of education

The aim of this course is to provide a student a comprehensive overview of programming methodology that can be useful in further independent research in quantum information

Course contents

Review and systematics of programming languages. Imperative and declarative programming. History and labor market. Programming environments.
Program structure in C ++, Python, Matlab.
Basic constructions. Variables, loops, conditional statements, functions, I / O operations, operators.
Object-oriented programming. Classes. Basic data structures. Array, list, heap, map, graph.

Code organization. Comments, headers, libraries, naming conventions. Programming Pragmatics. Programming styles. Version control systems. Doxygen.
Recursion. Dynamic programming. Basic algorithms. Searching, sorting, graph searching.
STL library in C ++. Design patterns. Processes and threads. Multi-threaded programming. Data Representations. XML. Sparse matrices. COO and CRS formats.
Functional programming.
Numerical Methods. Newton-Raphson method, Simpson method, Runge-Kutta method, matrix decompositions.
Numpy and scipy packages in Python. Matlab QETLAB package.
Linear and semi-definite programming. Solvers.
Computational models. Turing machine. Church's thesis. Computational and memory complexity of algorithms. Complexity classes P, NP, NPC, PSPACE. Compilation process and parameters. Debugging and profiling. Unit tests.
Code optimization techniques. Language interoperability. MEX files in Matlab. Extension modules in Python.
CISC and RISC architectures. Flynn taxonomy. MMX, SSE, AVX instruction sets. Programming on graphic cards. CUDA, PyTorch.
Virtual machines and emulators. Bytecode in Python. Assembler and low-level code optimization.
BPP, BQP, QMA complexity classes. Quantum programming languages

Bibliography of literature

Python3 Documentation, https://docs.python.org/3/index.html
GNU Octave Free Your Numbers – reference manual, https://octave.org/octave.pdf
C++ Reference, http://www.cplusplus.com/reference/
Matlab Reference Manual, https://www.mathworks.com/help/matlab/
W. Malina, P. Mironowicz, “Programowanie strukturalne. Trendy programowania”, PWN 2018 (in Polish).
P. Wróblewski, “Algorytmy, struktury danych i techniki programowania”, Helion 2015 (in Polish).
Material provided be the lecturer.

The learning outcomes (for the field of study and specialization)

K_W02
Student has in-depth knowledge of advanced mathematics, mathematical and computer methods necessary to solve physical problems of medium complexity and advanced in the area of quantum information and its technological aspects

K_W05
The student knows the theoretical basis of computational methods and information techniques used to model and simulate physical systems considered in the theory of quantum information

K_U02
The student can apply mathematical knowledge as well as mathematical and computer tools to formulate and solve problems within the framework of quantum information theory

K_U04
The student an find the necessary information in professional literature, both in databases and other sources; can recreate the reasoning or the course of an experiment described in the literature, taking into account the assumptions and approximations made

K_U07
The student can present the results of research (experimental, theoretical or numerical) in writing, orally, as a multimedia presentation or as a poster

Knowledge

W01: The student knows the components of programming languages C++, Python and Matlab, (K_W02, K_W05)
W02: The student knows basic algorithms and packages (K_W02, K_W05)
W03: The student knows good programming practices and basics of computer architecture (K_W02, K_W05)

Skills

U01: The student is able to write stand-alone code in C++, Python and Matlab designed to solve various types of scientific and numerical problems (K_U02).
U02: The student has skills necessary to properly design it, choose relevant tools and methods, validate the correctness of the code, find and overcome performance bottlenecks. (K_U02)
U03: The student should learn to efficiently get to know new techniques individual from relevant reference manuals. (K_U02)
U04: The student should learn how to find and ask about new sources of knowledge, cooperate on designing and writing a computer code, and present data in a way readable to other people (K_U04, K_U07)

Social competence

Contact

piotr.gnacinski@ug.edu.pl