COURSE DESCRIPTION CARD - SYLLABUS

Course name
Informative Methods in Physics and Technique [S1FT1>MIwFiT]

Course

Field of study
Technical Physics

Year/Semester
1/2

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Justyna Barańska
justyna.baranska@put.poznan.pl

Lecturers

dr inż. Justyna Barańska
justyna.baranska@put.poznan.pl

Prerequisites

Knowledge of physics, mathematics and computer science at the level after the first degree of education in the field of technical physics. Skill in basic use a Windows computer. Active attitude when solving problems, understanding the need to expand one"s competences

Course objective

1. Acquaintance of the students with the basics of the C ++ language that enables the creation of numerical programs that solve problems appearing in physics and technology. 2. Demonstrating the usefulness of the computer as a tool supporting the analysis and solving of simple physical and technical problems. 3. Developing student"s ability to use a computer independently to analyze simple problems in the field of physics and technology, in particular through the development and implementation of programs for simulating simple physical processes.

Course-related learning outcomes

Knowledge:
1. student, who has completed the course, is able to select and describe mathematical apparatus necessary to describe the laws of physics and solve problems, including: basic numerical algorithms used

in technical physics [k2_w01]
2. student, who has completed the course, is able to list and describe the structures and instructions of the selected programming environment supporting engineering calculations [k1_w05]

Skills:
1. student, who has completed the course, is able use mathematical knowledge to create computer models and write numerical algorithms in the field of technical physics [k2_u01]
2. student, who has completed the course, is able to correctly use the data structures and instructions of the selected programming environment and the learned numerical algorithms to solve a problem in the field of technical physics, perform visualization and computer simulation, and make a critical analysis of the obtained results [k1_u09, k1_u19]
3. student, who has completed the course, on the basis of the available documentation in english, can independently find additional information about data structures, instructions and available libraries [k1_u11].

Social competences:
1. student, who has completed the course, is able to work responsibly on a designated multi-threaded task, independently and in a team, [k2_k01]
2. student, who has completed the course, is able to notices the necessity of ethical use of computer software in accordance with its licenses [k1_k02].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
W01, W05 : : Assessment of knowledge demonstrated in written work on the grounds of scored points
Assessment of an individual oral presentation with the use of a computer program.
50.1%-70.0% (3)
70.1%-90.0% (4)
90.1%-100.0%(5)
U01, U09, U011, U019: Assessment of programming skills demonstrated in practice.
Assessment of an individual oral presentation with the use of a computer program.
Assessment of activity during Laboratory classes.
50.1%-70.0% (3)
70.1%-90.0% (4)
90.1%-100.0%(5)
K01, K02 Assessment of an individual oral presentation with the use of a computer program.

Programme content

A. Fundamentals
1. Introduction to Visual C ++, programs in text and graphical mode.
2. Support for standard input output in C and C ++.
3. Variables, constants. Defining a function.
4. Passing values to functions. The concept of indicator, reference, value.
5. Complex types: structures, typedef, pointers to functions.
6. Recursion - comparison of the implementation of the factorial (n) function in iterative and recursive terms.
7. Sorting algorithms.
8. Signal handling.
B. Object oriented programming
1. The concept of classes, constructors, properties, methods.
2. Static methods.
3. Inheritance, virtual methods.
4. Friendly classes, operator overloading.

Teaching methods

1. Lectures: multimedia presentation, presentation illustrated with examples given on the blackboard.
2. Laboratory classes: practical exercises, conducting, discussion, teamwork.

Bibliography

Basic
1. Jerzy Grębosz, "Symfonia C++: programowanie w języku C++ orientowane obiektowo", tom 1,2 i 3, Oficyna Kallimach, 2000.
2. BjarneStroustrup, Język C++", Wydawnictwa Naukowo-Techniczne, Warszawa 1998.
Additional

Breakdown of average student's workload

	Hours	ECTS
Total workload	90	3,00
Classes requiring direct contact with the teacher	50	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50