SYLLABUS – A COURSE DESCRIPTION

I. General information

1. Course name: Introduction to computer science and programming
2. Course code: ITCSP
3. Course type (compulsory or optional): compulsory
4. Study programme name: Language, Mind, Technology
5. Cycle of studies (1st or 2nd cycle of studies or full master's programme): 2nd cycle of studies
6. Educational profile (general academic profile or practical profile): general academic profile
7. Year of studies (if relevant): 1MA
8. Type of classes and number of contact hours (e.g. lectures: 15 hours; practical classes: 30 hours): practical classes: 30 hours
9. Number of ECTS credits: 3
10. Name, surname, academic degree/title of the course lecturer/other teaching staff: Robert Dyzman, M.Sc.Eng., robdyz@ext.amu.edu.pl
11. Language of classes: English
12. Online learning – yes (partly – online / fully – online) / no: no

II. Detailed information

1. Course aim (aims):
   - 1.1 Providing the basics of IT knowledge, terminology and computer workflow.
   - 1.2 Providing knowledge about creating flowcharts.
   - 1.3 Teaching the basics of C++ programming language
   - 1.4 Teaching the basics of Python programming language
   - 1.5 Providing knowledge about basic algorithms
   - 1.6 Providing knowledge about basic data structures

2. Pre-requisites in terms of knowledge, skills and social competences (if relevant):

3. Course learning outcomes (EU) in terms of knowledge, skills and social competences and their reference to study programme learning outcomes (EK):

Course learning outcome symbol (EU)	**On successful completion of this course, a student will be able to:**	**Reference to study programme learning outcomes (EK)**
ITCSP_01	knows the basic terminology. Has the ability to convert decimal numbers to binary and hexadecimal numbers and vice versa. Understands the computer's working pattern. Knows and understands the concept of low-level and high-level languages	K_W04, K_U18
ITCSP_02	can present example tasks in the form of flowcharts	K_W07
ITCSP_03	for l. C ++, knows data types and their characteristic features, can use them. Knows and is able to use control structures and functions. Can write a simple program in C ++. Can use environments DevC++ and , C++ Online Compiler https://www.onlinegdb.com/online_c++_compiler	K_W10, K_U15
ITCSP_04	for l. Python, knows data types and their characteristic features, can use them. Knows and is able to use control structures and functions. Can write a simple program in Python. Can use environments Spyder, Jupyter Notebook	K_W10, K_U15
ITCSP_05	understands the concept of an algorithm and the concept of computational complexity of an algorithm based Big-Oh notation. Can present a simple algorithm in the form of a flowchart and implement it in C ++ and Python. Can use recursion	K_W07, K_U14
ITCSP_06	understand the concept of data structure. It distinguishes elementary data structures, at least lists, stacks, queues	K_W09

4. Learning content with reference to course learning outcomes (EU)

Course learning content:	**Course learning outcome symbol (EU)**
Fundamentals of computer science. Terminology. Number systems	ITCSP_01
Flowcharts	ITCSP_02

Fundamentals of C++	ITCSP_03
Fundamentals of Python	ITCSP_04
Algorithms	ITCSP_05
Data structures	ITCSP_06

5. Reading list:

- Petzold Ch. (2000) Code: The Hidden Language of Computer Hardware and Software
- Benjamin Smith. (2021) C++: 3 in 1- Beginner's Guide+ Simple and Effective Tips and Tricks+ Advanced Guide to Learn C++ Programming Effectively
- N.Wirth (2000) Algorithms + Data Structures = Programs
- Allen B. Downey: Think Python: How to Think Like a Computer Scientist https://www.greenteapress.com/thinkpython/html/index.html

III. Additional information

1. Teaching and learning methods and activities to enable students to achieve the intended course learning outcomes (please indicate the appropriate methods and activities with a tick and/or suggest different methods)

Teaching and learning methods and activities	**X**
Lecture with a multimedia presentation	X
Interactive lecture	
Problem – based lecture	
Discussions	X
Text-based work	
Case study work	
Problem-based learning	
Educational simulation/game	
Task – solving learning (eg. calculation, artistic, practical tasks)	
Experiential work	X
Laboratory work	
Scientific inquiry method	
Workshop method	
Project work	
Demonstration and observation	
Sound and/or video demonstration	
Creative methods (eg. brainstorming, SWOT analysis, decision tree method, snowball technique, concept maps)	
Group work	X
Other (please specify) -	
…	

2. Assessment methods to test if learning outcomes have been achieved (please indicate with a tick the appropriate methods for each LO and/or suggest different methods)

Assessment methods	**Course learning outcome symbol**					
	ITCS P_01	ITCS P_02	ITCS P_03	ITCS P_04	ITCS P_05	ITCS P_06
Written exam						
Oral exam						
Open book exam						
Written test						
Oral test						
Multiple choice test	X	X	X	X	X	X
Project	X	X	X	X	X	X

Essay						
Report						
Individual presentation						
Practical exam (performance observation)						
Portfolio						
Other (please specify) -						
…						

3. Student workload and ECTS credits

Activity types		**Mean number of hours spent on each activity type**
Contact hours with the teacher as specified in the study programme		30
Independent study*	Preparation for classes	20
	Reading for classes	15
	Essay / report / presentation / demonstration preparation, etc.	
	Project preparation	10
	Term paper preparation	
	Exam preparation	
	Other (please specify) -	
	…	
Total hours		75
Total ECTS credits for the course		3

* please indicate the appropriate activity types and/or suggest different activities

4. Assessment criteria in accordance with AMU in Poznan's grading system:

Very good (bdb; 5,0): student knows and understands the concepts very well, can successfully apply them
Good plus (+db; 4,5): student knows and understands the concepts very well, can successfully apply them, but makes small errors
Good (db; 4,0): student knows and understands the concepts well, can successfully apply them, but makes occasional errors
Satisfactory plus (+dst; 3,5): student knows and understands the concepts on a basic level, can apply them on a satisfactory level, but makes errors
Satisfactory (dst; 3,0): student knows and understands the concepts on a basic level, can apply them on a basic level, but makes errors
Unsatisfactory (ndst; 2,0): student does not know or understand the concepts and cannot apply them without making gross errors