Programming basics I

Aim of the course	Aim of the course is to introduce students with structured problem solving, basic principles and application of algorithms and object-oriented programming. This course will develop the ability for students to deconstruct programming problems and construct their own solutions for them. Result of this course will be a strong springboard to students for learning more programming languages and deepen their understanding about programming in the future.
Methods	This course will consist of theoretical lectures merged with live coding examples and discussions; code reading tasks; practical coding sessions with examples and explanations; coding tests; extensive set of homework; course work and exam. Course also has video lectures and code examples available in video format.
Course description	This course is an introduction to the design and implemention of programming languages (mainly Java). From the design point of view, we will study language features as tools for expressing algorithms. Course will start from absolute basics without the need of any prior knowledge of programming. Topics covered will include: * Understanding of how programming works * Problem deconstruction and use of flowcharts * Synthax and basic variable and operations * Method definition and calling * Class and object definition * Data input/output, error catching * Good programming practices * Understanding of clean, well structured and defensive code
Course work	Course work will include a solution of chosen programming problem, descriptive documentation and demonstration in class. More info in the Moodle
Exam	Exam consists of three theoretical questions, two code reading tasks and four coding tasks. Students with at least 8.0 GPA, no grade less than 7 and well presented course work will be granted an automatic pass in the exam
Interactive methods	Course includes discussion lectures about code read, construction of algorithms etc. No lecture consists of only theoretical presentations. Practical coding tasks with interaction with the lecturer and live coding all together are also examples of interactive methods used in this course Course also has consultations with option to rewrite tests, receive more detailed explanation about lecture topics as well as 1-to-1 discussions

Grades	Final grade is calculated of:
	15% - homework;
	20% - tests;
	10% - practical tasks in class;
	25% - course work;
	30% - exam;
	Exam is calculated as THEORY x 0.2 + CODE_READ x 0.3 + CODING x 0.5

List of lectures

Topic	Description	Aim	Result	Test	Homework
Introduction of programming, Java, OOP and reason behind programming languages	Explanation of how things work behind the curtain, why Java is chosen as the base of the course and why programming was created and what purpose it serves and will serve in the future How to behave in lectures, how to study at home.	Break the ice	Broaded the view of students and give understanding and motivation to study this particaular course		
Simple data types, variables, arrays and their differences in programming languages	Definition and usage of the simple data types used in Java and the difference of simple data type and variable implementation in different programming languages. Explanation, usage and restrictions of arrays and different types of arrays	Learn about simple data types and arrays and their usage and synthax in Java	Be able to use variables of simple data types and understand differences between types and their application		Coding task on topics covered
Operations and operators	Usage of arithmetical operations, comparing, logical comparing, assignment, conditional operations; order of operations Different types of branching and looping operators	Learn to use simple variables and arrays in branching operations and loops; learn the synthax of different basic operations and application of branching and looping. Learn to use mentioned tools for basic mathematical problem solving	Be able to create code that branches and loops where necessary. Be able to code synthactically correct loops and branches. Learn how to use mentioned tools to deconstruct simple algorithmic problems and solve them with code.		Coding task on topics covered
Methods	Definition and usage of static methods; their synthax, data types, accessibility types; argument passing and receiving, returning data types, method calls; good method code structure.	Learn to use methods to create non-repetitive and qualitative programming code. Learn about how to create dynamic methods to be able to use one method in multiple situations	Be able to create code using the DRY principle, understand method calls and return of the data. Be able to apply previously learned tools to methods.		Coding task on topics covered
Keywords *break, continue, return*	Usage of different keywords in methods and loops.	Learn how to differentiate and use keywords [break, continue, return] in methods and loops	Be able to write code using keywords [break, continue, return] making code faster, more readable and suitable to specific situations	Code reading test about basic operators and operations	
String operations	Different set of String operations and basic explanation of RegExes	Learn about how to apply default String operations and manipulate text	Be able to correctly choose from and use wide variety of String operations to manipulate text programmatically		
Data input from keyboard and Error handling					Coding task on topics covered
Classes and objects	Definition of custom class, it's parameters, constructor, static and dynamic class methods, creation of objects with custom constructor calls; parameter and method calls for objects. Object variables and references. Different access levels	Learn the basics of object creation and their usage, difference between objects of the same class and their carried values	Be able to define new class and create method of the type. Differentiate dynamic and static methods in the class. Understand the usage of the class and apply that in problem solving. Understand good coding principles in basic object oriented programs.	Coding test about String operations	Coding task on topics covered
Dynamic collections	Definition of Lists, Sets, Maps and differences between the types and different subtypes of one type. Differences between dynamic lists and static arrays and their application	Learn about different dynamic lists and their characteristics as well as how and when to use them	Be able to choose optimal dynamic list in any given situation, differentiate pros and cons of each list type in different situations. Understand when static arrays are better choice than dynamic lists.	Coding test about methods	
File processing	How file reading and writing operations work. Different approaches. What different file types consist of. Write code that can read and write into .csv and .txt files.	Learn about how file reading/writing is done. Learn about how to use appropriate file for data storing, what are different approaches of file reading	Be able to save data in file and use that data in the future by reading the contents. Be able to apply mentioned ability to problemsolving situations.		Coding task on topics covered
Good quality programming principles	Basics of defensive programming, DRY code principles, differences of different levels of code (bad, average, pro). Review of different codes	Learn how to write valuable, clean code that uses defensive programming principles. How to maintain high code quality in different iterations	Be able to plan for and write defensive and high quality code despite the programming language chosen	Code reading test about methods	Analyzing own code
Presentation of course work	10 minutes for each student explaining and demonstrating their course work	Broaden the view of different applications of the code by peers. Learn about different approaches of using code in problemsolving			
Exam	3 theoretical questions, 2 code reading tasks, 4 coding tasks				