Óbuda University John von Neumann Faculty of Informatics	Institute of Software Engineering

Subject name: Software Design and Development I.
Subject code: NIXSF1EBNE
Credits: 6

Computer science BSc

Full-time, 2023/24 autumn semester

Subject lecturers: Dr. László Csink, Dániel Kiss				
Prerequisites:	-			
Weekly hours:	Lecture: 3	Seminar: 0	Lab hours: 3	Consultation: 0
Way of assessment:	Examination			

Course description

Goal: Students will learn the rudiments and main methods of OOP, as well as get an introduction to a modern OO programming language.

Topics: The main competences are Algorithm design, control structures. Description of algorithms. Simple and Comopund Basic Programs. Combining Basic Programs. The OOP paradigm: objects, classes, encapsulation, hiding, inheritance, polymorphism. Sorting and searching. Sets. Recursion. Mergesort and Quicksort. Elementary number theoretical algorithms.

Lecture schedule

Week	Topic
1.	The basics of algorithms
2.	Simple Basic Programs
3.	Compund Basic Programs
4.	Value and reference types
5.	Combining Basic Programs
6.	Sorting
7.	Searching algorithms
8.	Sets, representation, set operations
9.	Recursion
10.	Divide and Conquer algorithms: Mergesort and Quicksort
11.	*Rector's holiday*
12.	Dynamic Programming
13.	Number Theoretical Algorithms
14.	Consultation

Midterm requirements

Students must write two midterm exams (computer programming tasks) on weeks 7 and 13 in lab lesson time. Both tests are expected to be at least 50%. If a student has not written either of the tests, or its result is less than 50%, the test must be rewritten in the last week. The result of the test will be the result of the rewriting. If a student missed both tests or missed one of the tests and the other's result is less than 50% or has written both tests but both results are weaker than 50%, the signature can be obtained only at the so-called signature test that will take place in the examination period. Even if both tests are better than 50%, students are allowed to rewrite the worse test. The final result of the test will be the result of the rewriting (even if it is worse than the previous result).

Students will get a home project on week 4, that must be handed in until December 3. It is possible to get a one-week extension of this deadline, but in this case, a special fee must be paid. The specifications of the requirements concerning the home projects will be uploaded to the Moodle.

Midterm exams	
Week	Topic
7	First midterm exam on implementing algorithms in C#
13	Second midterm exam on OOP and algorithms in C#
14	Midterm exam retake (optional)

Midterm grade calculation methods

At the end of the term period, the student can obtain a signature.

To obtain the signature, students must have both midterm exams completed at least 50% (see above) and have a submitted and successfully defended home project by the end of the term period.

If a student misses more lab lessons that 30% of all classes, or has not submit or defend the home project by the required deadline, the signature will be banned (so it cannot be completed on the signature retake exam).

In case the signature is not banned, but the student did not completed both midterm exams succesfully, the signature can be obtained on the signature retake exam.

Method of replacement

Students are expected to write both midterm tests with a result not lower than 50% each. At the last week one of tests can be rewritten, if necessary. If one has to write the signature test, must achieve not less than 50%. In case of success, the midterm activity will be evaluated 50% even if your signature test result is higher.

Type of exam

The exam will have a written part and an oral part. To pass the written part, you have to complete an entry test in the Moodle system. If you fail the written part, you cannot continue the oral part. Your final grade will be determined by taking your lab points as well as your written and oral part results into account, however, the final grade is not simply the arithmetical mean of those grades.

Exam grade calculation methods

0-49%: Insufficient (1)
50-61%: Sufficient (2)
62-73%: Satisfactory (3)
74-85%: Good (4)
86-100%: Excellent (5)

References
Obligatory:
Al Aho and Jeff Ullman: Foundations of Computer Science (http://infolab.stanford.edu/ ullman/focs.html) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein: Introduction to Algorithms, The MIT Press; 3rd edition (July 31, 2009). Online teaching materials available on the e-learning site of the course