**COURSE OUTLINE**

**Course: CS 349 Compilers**

|  |
| --- |
| **Lecturer:** Denis Sinanaj, Master of Science  **Credits:** 8  **Weekly load:** 3 lectures/ 2 seminars  **Session:** Spring  **Study program:** Bachelor in computer Science  **Code:** CS 349  **Email:** denissinanaj@univlora.edu.al |

|  |
| --- |
| **SUMMARY AND LEARNING RESULTS**  The course provides basic knowledge of compilers, functions, processes, phases of a compiler and their components. The subject has a broad aspect and will try to acquaint students with the basics of compilers, their importance in programming, the transformations that the code encounters when being translated it by the compiler. The subject addresses in more detail the stages in which a source program passes to be translated from the compiler into a comprehensible and executable language by machines.  **Objectives:**  Student to recognize the characteristics of compilers.  Familiarize with their structure. |

|  |
| --- |
| **BASIC CONCEPTS:**   1. What compilers are and their stages. 2. Lexical and syntax analysis. 3. Intermediate Code. 4. Register allocation. 5. Analysis and optimization. 6. Functional calls and memory management. |

|  |
| --- |
| **COURSE CONTENT**  **Topics at lectures:**  **I** - Introduction to compilers and stages of a compiler.  **II** - Lexical analysis.  **III** - Syntax analysis.  **IV** -Symbol table, interpreters and code control.  **V** - Syntax-oriented translation, schemes, and applications of this translation.  **VI** - Intermediate code generation. Intermediate language and data translation.  **VII** - Register allocation, types of allocation  **VIII** - Executing environments in time, space organization, Garbage collection and based on traces  **IX** - Code generation, phases, generated code optimization and dynamic programming.  **X** - Optimization, analysis of data flow, the basis of this analysis, elimination and graphs.  **XI** - Optimization about parallelisms and location.  **XII** - Interprocedural analysis, types of analysis and algorithms of this analysis.  **XIII** - Function calls and memory management.  **XIV** - Review and examples.  **XV** - Final project.  **Topics at seminars:**  **I -** Introduction to Compilers and compiler phases.  **II -** Lexical analysis.  **III -** Syntax analysis.  **IV -** Table of symbols, interpreters and code control.  **V -** Syntax oriented translation, schemes and translation application.  **VI** - Intermediate code generation. Intermediate language and data translation.  **VII** - Register allocation, types of allocation.  **VIII -** Executing environments in time, space organization, Garbage collection and based on traces.  **IX -** Code generation, phases, generated code optimization and dynamic programming.  **X -** Optimization, analysis of data flow.  **XI - B**asics of data flow analysis, elimination and graphs.  **XII -** Optimization about parallelisms and location.  **XIII -** Interprocedural analysis, types of analysis.  **XIV -** Algorithms of interprocedural analysis.  **XV -** Functional calls and memory management. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ASSESSMENT FORMS**  **ATTENDANCE:** The student, which results in less than 75% attendance for the period of each partial examination, the period for which the test is to be attended cannot access the respective exam and will be evaluated by M(Absense). If the student has attended the course, but does not appear in the next exam, will be evaluated NP (Not Present).  **CONTINOUS CONTROL:**  The course will be evaluated on the basis of a partial examination, annual assessment and final exam:   |  |  |  | | --- | --- | --- | | **Partial exam** | **Annual assessment** | **Final Exam** | | 30 % | 10 % | 60% |   The grade rating is based on the conversion of the total assessment in%, grades 5-10 progressively 41 to 100% |

|  |
| --- |
| **LITERATURE**  **a) Mandatory Basic Literature**:  [1]. Compilers, Principles, Techniques, & Tools. Second Edition. Alfred V. Aho, Monica S. Lam, Monica S. Lam, Jeffrey D. Ullman. 978-0-321-48681-3, 2007.  **b) Auxilliary literature**:  [2] Basics of Compiler Design, Torben Ægidius Mogensen, ISBN 978-87-993154-0-6, 2010.  **Communication:**  Homework exercises, course assignments and any other announcements will be given in the classroom and / or at the official address of Vlora University "Ismail Qemali" on the Internet: ww.univlora.edu.al. or at the email of lecturer Denis Sinanaj (denis.sinanaj@univlora.edu.al).  **Honesty code:**  Copying from one another to exams, course assignments, etc. is not allowed. Breaking this rule will be accompanied by punitive measures that go up to university exclusion.  **Mandatory rules on the course:**  It is not allowed to use mobile phones and smoking at the auditorium. |

**Approved by**

**Head of computer science department**

**Dr. Eljona PROKO**