

SYLLABUS: ALGORITHM, PROGRAMMING 1

Date / Revision 15 August 2017 / Rev.02 15.08.17 /MaS
Faculty Engineering and Lifesciences
Study Programs AVE, AUE, COS, MEE, INE, ELE, MTE, FTE, CHE, BME

SUBJECT: Algorithm, Programming 1

1 Basic Information

1.01	Subject Name	Algorithm, Programming 1
1.02	Semester	1
1.03	Level	1
1.04	SKS	3
1.05	Mandatory / Curriculum	F-06
1.06	Subject Code	PROG
1.07	Subject Code	ENG-F-PROG-116
1.08	Year	2017 (7)
1.09	Quality Control	Final Test, see evaluation
1.10	Limitations	Min 12 and Max 32 students in one class
1.11	Combined with	AVE, AUE, COS, MEE, INE, ELE, MTE, FTE, CHE, BME
1.12	Perquisite	None
1.13	Responsible	Mr. DR. Prianggada Indra Tanaya, MMfg / Ir. Neno Suseno, MSc.
1.14	Revision	15-08-2017/MaS

2 Description of Subject

The subject introduces students with the knowledge and skill of algorithm, programming language using C/C++, and data structure. The algorithm will be introduced by using daily step-by-step algorithm of well known daily life activity, idea of how to move robots, and others. Using an easy to understand mnemonics, slightly and easy migration from daily language to almost executable code is introduced. Next, that will be transferred into the syntax of C and/or C++ programming language syntax. Student introduces to flowcharting, Unified Modeling Language (UML) to be able to describe the sequence and architecture of a program. Later, software IDE (Interactive Development Environment) is used to show that the programs created and developed are executable on computer.

3 Objectives

- This course introduces students to the analysis and design of computer algorithms using algorithm and programming with C++
- to analyze the asymptotic performance of algorithms.
- to familiarize with major algorithms and data structures.
- to apply important algorithmic design paradigms and methods of analysis engineering problems

4 Competency

After having the course, students are expected to:

- Explain what is algorithm in general,
- Develop an algorithm for solving engineering problems,
- Use C programming language for solving engineering problems,
- Build and develop a program with data structure,
- Apply the function for solving engineering and computing problems,
- Debug, compile and execute a program, and
- Use one type of C/C++ programming IDE tool to develop/build a program based on algorithm to solve engineering problems.

5 Learning Approach / Methodology

- Lectures/ Class contact (time-tabled) supplemented with interactive questions and answers;
- Circuit simulation using Electronic Workbench Software;
- Tutorial/Laboratory/Practice Classes: preview of materials, revision and/or reports writing;
- Student Study Effort: homework/assignment; preparation for test/quizzes/ examination.

6 Evaluation

5.1	Absence maximum	25%
5.2	Participation in Discussion	05 Points
5.3	Homework / Classwork	05 Points
5.4	Presentation /Simulation	10 Poin
5.5	Daily Quiz	20 Points
5.6	Final Examination	60 Points
	Total	100 Points

7 Text Book and Reference

1	<p>Main Text Book: <i>“C++ How to program, 9th-Edition”</i>, Authors: Deitel P. and Deitel H., Publisher: Pearson Higher Education, ISBN: 9780273793298</p>
2	<p>Supplement Textbooks:</p> <ul style="list-style-type: none"> • <i>“C++ Programming: Program Design Including Data Structures, Fifth Edition”</i>, Authors: D.S. Malik, Publisher: Course Technology – CENGAGE Learning, ISBN: 978-0-538-79809-2 • <i>“C++ Programming: From Problem Analysis to Program Design, Fifth Edition”</i>, Authors: D.S. Malik, Publisher: Course Technology – CENGAGE Learning, ISBN: 978-0-538-79808-2 • <i>“C++ Programming for the Absolute Beginner, Second Edition”</i>, Author: Mark Lee, Publisher: Course Technology – CENGAGE Learning, ISBN: 978-1-59863-875-2

8 Content / Topics of Lecture

Week	Content/ Topics of Lecturing	Text Book	Remark
1-2	<p>Intro to Course Structure, Quizzes, Exam and Grading Policy:</p> <ul style="list-style-type: none"> • Course Structure • Quizzes, Homework, Assessment, and Participation <p>Introduction to Computers and C++ Program:</p> <ul style="list-style-type: none"> • Introduction • Computers and the Internet in Industry and Research • Hardware and Software • Data Hierarchy • Machine-, Assembly- and High Level - Languages • C++ Programming Languages • Introduction to Object Technology • Typical C++ Development Environment • Test-Driving a C++ Application • Operating Systems • The Internet and World Wide Web • C++ and the Open Source Boost Libraries • Keeping Up to Date with Information Technologies 	Ch-1	
3	<p>Introduction to C++ Programming; Input / Output and Operators:</p> <ul style="list-style-type: none"> • First program in C++: Printing a Line of Text • Modifying our First C++ Program • Another C++ Program: Adding Integers; • Memory Concepts; • Arithmetic; • Decision Making: Equality and Relational Operators 	Ch-2	

4-5	<p>Introducing to Classes, Objects and Strings:</p> <ul style="list-style-type: none"> • Defining a Class with Member Function • Defining a Member Function with a Parameter • Data Members, <i>set</i> Member Functions and <i>get</i> Member Function • Initializing Objects with Constructors • Placing a Class in a Separate File for Reusability • Separating Interface from Implementation • Validating Data with set Functions 	Ch-3	
6	<p>Control Statements: Part I; Assignment, ++ and – Operators:</p> <ul style="list-style-type: none"> • Algorithms and Pseudocode • Control Structures: <ul style="list-style-type: none"> - if Selection Statement - if...else Double Selection Statement - while Repetition Statement • Formulating Algorithms: Counter Controlled Repetition • Formulating Algorithms: Sentinel Controlled Repetition • Formulating Algorithms: Nested Controlled Repetition • Assignment Operators: Increment and Decrement Operators 	Ch-4	
7	<p>Control Statements: Part I; Logical Operators:</p> <ul style="list-style-type: none"> • Essential of Counter Controlled Repetition • for Repetition Statement • Example Using the for Statement • do ... while Repetition Statement • switch Multiple Selection Statement • break and continue Statement • Logical Operators • Confusing the Equality (==) and Assignment (=) Operators • Structured Programming Summary • Wrap - Up 	Ch-5	
8	MIDTERM SEMESTER BREAK		
9	<p>Function and an Introduction to Recursion:</p> <ul style="list-style-type: none"> • Program Components in C++ • Math Library Functions • Function Definitions with Multiple Parameters • Function Prototypes and Argument Coercion • C++ Standard Library Headers • Case Study • C++11 Random Numbers • Storage Classes and Storage Duration • Function Call Stack and Activation Records • Functions with Empty Parameter Lists • References and Reference Parameters, Default Arguments • Unary Scope Resolution Operator • Function Overloading, Function Templates, Recursion • Example Using Recursion: Fibonacci Series • Recursion vs. Iteration 	Ch-6	

10	<p>Class Templates array and vector; Catching Exceptions</p> <ul style="list-style-type: none"> • Arrays: Declaring arrays, Examples Using arrays • Range-Based for Statement • Case Study: Class GradeBook Using an array to Store Grades • Sorting and Searching arrays • Multidimensional arrays • Case Study: Class GradeBook Using a Two-Dimensional array • Introduction to C++ Standard Library Class Template vector 	Ch-7	
11-12	<p>Pointers:</p> <ul style="list-style-type: none"> • Pointer Variable Declarations and Initialization • Pointer Operators • Pass-by-Reference with Pointers • Built-In Arrays • Using const with Pointers • sizeof Operator • Pointer Expressions and Pointer Arithmetic • Relationship Between Pointers and Built-In Arrays • Pointer-Based Strings 	Ch-8	
13	<p>Classes : A Deeper Look; Throwing Exceptions</p> <ul style="list-style-type: none"> • Time Class Case Study • Class Scope and Accessing Class Members • Access Functions and Utility Functions • Time Class Case Study: Constructors with Default Arguments • Destructors • When Constructors and Destructors Are Called • Time Class Case Study: A Subtle Trap—Returning a Reference or a Pointer to a private Data Member • Default Memberwise Assignment • const Objects and const Member Functions • Composition: Objects as Members of Classes • friend Functions and friend Classes • Using the this Pointer • static Class Members 	Ch-9	
14	<p>Operator Overloading; Class string</p> <ul style="list-style-type: none"> • Using the Overloaded Operators of Standard Library Class string • Fundamentals of Operator Overloading • Case Study: • Operators as Member vs. Non-Member Functions • Converting Between Types • explicit Constructors and Conversion Operators • Overloading the Function Call Operator () 	Ch-10	
15	<p>Object Oriented Programming</p> <ul style="list-style-type: none"> • Base Classes and Derived Classes • Relationship between Base and Derived Classes • Constructors and Destructors in Derived Classes • public, protected and private Inheritance • Software Engineering with Inheritance 	Ch-11	
16	Final Examination		