HANDBOOK

BACHELOR OF INFORMATICS PROGRAM
DEPARTMENT OF INFORMATICS
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

Module name	**Software Construction**
Module level	Undergraduate
Code	IF184974
Courses (if applicable)	**Software Construction**
Semester	7
Contact person	
Lecturer	Rizky Januar Akbar, S.Kom., M.Eng.
Language	Bahasa Indonesia and English
Relation to curriculum	1. Undergraduate degree program; optional; 7th semester. 2. International undergraduate program; optional; 7th semester.
Type of teaching, contact hours	1. Undergraduate degree program: lectures, < 60 students, 2. International undergraduate program: lectures, < 40 students
Workload	1. Lectures: 3 x 50 = 150 minutes (2 hours 30 minutes) per week. 2. Exercises and Assignments: 3 x 60 = 180 minutes (3 hours) per week. 3. Private study: 3 x 60 = 180 minutes (3 hours) per week.
Credit points	3 credit points (sks).
Requirements according to the examination	A student must have attended at least 80% of the lectures to sit in the exams.

regulations		
Mandatory prerequisites	Analysis and Design of Information Systems and Software Design (taken)	
Learning outcomes and their corresponding PLOs	After completing this module, a student is expected to:	
	CO1 Students are able to explain essential and accidental complexities in software development.	
	CO2 Students are able to explain the stages in the software construction phase.	

	CO3 Students are able to translate the detailed design of the software into program code.	
	CO4 Students are able to determine the platform, language, and standard required according to the type of software to be built.	
	CO5 Students are able to build software using best practices in the coding, debugging, testing, and integration processes	
	CO6 Students are able to produce high quality program code.	
	CO7 Students are able to make program code improvements.	
	CO8 Students are able to collaborate and integrate software.	
Content	Knowledge: • Mastering the concepts and principles: design and development software with planning methods, requirements engineering, design, implement, test, and launch standards and scientific, and produces software products that fulfil various technical and managerial quality parameters, and power use and master the concepts and principles: programming simple in general programming languages and languages object-oriented programming, creation of web applications and applications desktop, database creation is simple to complete problems in the context of software development general; • Mastering the principles of making an algorithm and various programming language concepts; Specific Skill: • Able to analyze, design and build software using software engineering process principles to produce software that meets both technical and managerial quality; • Able to design and analyze algorithms to solve problems effectively and efficiently based on strong programming principles, and be able to apply programming models that underlie various existing programming languages, and be able to choose a programming language to produce suitable applications;	
Study and examination requirements and forms of examination	Mid-terms examination and Final examination.	
Media employed	LCD, whiteboard, websites, books (as references), etc.	
Assessments and Evaluation		

Reading List	McConnell, S.Code Complete: A Practical Handbook of Software Construction, 2nd Edition. Redmond, Wash: Microsoft Press, 2004. Fowler, Martin, and Kent Beck. Refactoring: Improving the Design of Existing Code. Reading, MA: Addison-Wesley, 1999. Martin, Robert C., and Micah Martin. Agile Principles, Patterns, and Practices in C♯. Upper Saddle River, NJ: Prentice Hall, 2007. Brooks, Frederick P. The Mythical Man-month Essays on Software Engineering. - Anniversary Ed. Reading, Mass.: Addison-Wesley Pub., 1995. Gamma, Erich. Design Patterns: Elements of Reusable Object-oriented Software. Reading, Mass.: Addison-Wesley, 1995.