HANDBOOK

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

Module name	**Framework Based Programming**
Module level	Undergraduate
Code	IF184605
Courses (if applicable)	**Framework Based Programming**
Semester	6
Contact person	
Lecturer	
Language	Bahasa Indonesia and English
Relation to curriculum	1. Undergraduate degree program; mandatory; 6th or 8th semester. 2. International undergraduate program; mandatory; 6th or 8th 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	Object Oriented Programming	
Learning outcomes and their corresponding PLOs	After completing this module, a student is expected to:	
	CO1 Students are able to explain the basic concepts of framework design.	
	CO2 Students are able to explain the differences between traditional software development and framework-based software development.	

	CO3 Students are able to explain several types of frameworks in different domains	
	CO4 Students are able to identify the advantages and disadvantages of using the framework	
	CO5 Students are able to identify a framework in accordance with the problems and / or needs of the user	
	CO6 Students are able to identify limitations in framework-based software development.	
	CO7 Students are able to design software designs by considering the framework.	
	CO8 Students are able to implement software using several frameworks	
	CO9 Students are able to add new functionality to a framework (extension).	
Content	Knowledge: Mastering the principles of making an algorithm and various programming language concepts. Specific Skill: Be 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	Cwalina, K., Abrams, B., “Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries 2nd Edition”, Addison- Wesley, Boston, 2008 McConnell, S., “Code Complete: A Practical Handbook of Software Construction, 2nd Edition”, Microsoft Press, Redmond, 2004