# TKU211131

Fundamentals of Programming

Pemrograman Dasar

## BASIC INFORMATION

| | |
|---|---|
| **Course Credit** | 3 / 150 minutes per Week |
| **Course Type** | Required |
| **Course Classification** | Engineering Topics |
| **Prerequisites** | - |

## STUDENT AND LEARNING OUTCOMES

### Covered Student Outcomes

Fundamental and Engineering Knowledge (KP.1)

Development of Engineering Solution (KP.2)

Modern Tools Utilization (SK.1)

### Learning Outcomes

**LO1** Student are able to explain the concept of programming, including syntax, error handling and file managements.

**LO2** Students are able to develop procedural pradigm programming.

**LO3** Student are able to utilize various data types and basic data structures to develop the programs.

**LO4** Students are able to implement effective and error-free programs.

## COURSE DESCRIPTION

This course will discuss about program development steps, ranging from defining problem, determining program input & output, and determining steps by utilizing operator and operands, data types, structure, programming control. This course also elaborates programming strategies and modularity.

## TOPICS

**PART 0 : MOTIVATION**

**1. Computer, People, and Programming**

1.1 Introduction
1.2 Software
1.3 People
1.4 Computer are everywhere
1.5 Ideal for Programmer
1.6 History, ideals and professionalism
1.7 Programming Language History Overview

**PART I : THE BASIC**

**2. Hello World!**

2.1 Program

2.2 The classic first program
2.3 Compilation
2.4 Linking
2.5 Programming Environments

**3. Object, Types, and Values**
3.1 Input
3.2 Variables
3.3 Input and type
3.4 Operations and Operators
3.5 Assignment and initialization
3.6 Composite assignment operators
3.7 Names
3.8 Types and Objects
3.9 Type safety (Safe & Unsafe conversions)

**4. Computation**
4.1 Computation
4.2 Objective and tools
4.3 Expressions
4.4 Statements
4.5 Functions

**5. Error**
5.1 Introduction
5.2 Sources of errors
5.3 Compile-time error
5.4 Link-time error
5.5 Run-time errors
5.6 Exceptions
5.7 Logic errors
5.8 Estimation
5.9 Debugging
5.10 Pre- and Post-conditions
5.11 Testing

**6. Writing a Program**
6.1 Thinking about the problem
6.2 Grammar & Code
6.3 Program Structure

**7. Completing a Program**
7.1 Introduction
7.2 Input and Output
7.3 Error handling
7.4 Negative numbers
7.5 Remainder
7.6 Cleaning up the code
7.7 Recovering from errors
7.8 Variables

**8. Functions**
8.1 Declarations and Definitions
8.2 Header files

8.3 Scope
8.4 Function call and return
8.5 Order of evaluations
8.6 Namespaces

### 9. Classes

9.1 User-defined types
9.2 Classes and members
9.3 Interface and implementation
9.4 Evolving a class
9.5 Enumerations
9.6 Operator Overloading
9.7 Class Interfaces

## PART II : INPUT AND OUTPUT

### 10. Input and Output Stream

10.1 Input and Output
10.2 The I/O stream model
10.3 Files
10.4 Opening a file
10.5 Reading and writing a file
10.6 I/O error handling
10.7 Reading a single value
10.8 User-defined output & input operators
10.9 A standard input loop
10.10 Reading a structured file

### 11. Customizing Input and Output

11.1 Regularity and irregularity
11.2 Output formatting
11.3 File opening and positioning
11.4 String streams
11.5 Line-oriended input
11.6 Character classification
11.7 Using nonstrandard separator

### 12. Testing

12.1. Introduction to Testing
12.2. Testing Procedure
12.3. Design for testing
12.4. Debugging
12.5. Performance

## REFERENCES

[1] Programming Principle and Practice Using C++ 2nd Ed. (Bjarne Stroustrup)
[2] The C++ Programming Language 4th Ed. (Bjarne Stroustrup)