TIN62109 – Computer Programming (3/P)

Description

The course is about computer programming, focusing on the introduction to Python. It is a beginning course (not for experienced programmers) in which students can learn and practice coding. The course includes video content, practice labs, and coding projects.

The course content is presented in three units.

- Unit 1 focuses on Python language basics, introducing data types, variables, input, functions, operators, conditional statements, loops, and incrementing. Students will be introduced to the basic structure of the Python 3 language and be ready to take Unit 2 as well as other beginner courses.
- Unit 2 focuses on Python data structures such as strings, lists, and range sequences, as well as methods for working with these structures. Students will be introduced to data structures and files in Python 3 and be ready for more advanced Python learning.
- Unit 3 focuses on using Python to develop sustainable code. Students will be introduced to data structures and files in Python 3 and be ready for more advanced Python learning.

The course content is inline to Microsoft Certification in the Data Science Certification route.

Course Timing

This course will be delivered over approximately 20 weeks, at a rate of one module per week. An average beginner student can complete the course instruction and core lab assignments in 3-4 hours per module. Additional study time can be allocated to exploring suggested resources and experimenting with knowledge gained while completing the practice tasks and labs.

Grading

Module assessments (Quizzes, Mid and Final semester assessments)	50%
Module coding assignments	20%
Final coding assignment (Projects)	30%

Learning Management Systems

Students must enrol themselves in iLearn (https://eng.ilearn.unand.ac.id/) and course groups in Microsoft Teams (http://portal.office.com/). Microsoft PowerPoint slides and other course-related files will be provided in both systems.

Tool

For the use of ease and implementation, this course will focus on using Visual Studio Code (https://code.visualstudio.com/) Integrated Development Environment (IDE). Visual Studio Code is a lightweight but powerful source code editor which runs on the desktop and is available for Windows, macOS, Linux, and Chromebook. Course modules with the practices will use the Visual Studio Code.

Course Content

Unit 1 (Five Weeks)

Overview

- Module 1: Python and Jupyter Basics
- Module 2: Functions
- Module 3: Conditionals
- Module 4: Nesting and Loops
- Module 5: Final Evaluation

Objectives

- Learn basic programming concepts
- Perform online exercises in the Jupyter Notebook development environment
- Develop and run programs in Jupyter Notebooks
- Utilize concepts such as data types, variables, and user input
- Build functions, use operators, use conditional statements
- Build while loops, increment variables
- Troubleshoot errors
- Complete coding assignments related to the content of each module

Module #	Modules	Sections
1	Python and Jupyter Basics	1.1: Starting Jupyter Notebooks 1.2: Types and Variables 1.3: Type Function 1.4: Addition and Errors 1.5: ASCII Art 1.6: Input 1.7: Print Formatting 1.8: Quote Display and Boolean 1.9: String Formatting and the 'in' Keyword **Module 1 Labs: Basics Practice** Module 1 Project
2	Functions	2.1: Simple Functions 2.2: Function Return and Multi-parameters 2.3: Sequence **Module 2 Lab: Functions Practice** Module 2 Project
3	Conditionals	3.1: Conditionals: Boolean Strings 3.2: Conditionals: Comparison Operators 3.3: String Comparison 3.4: Conditionals, elif, and Casting 3.5: Conditionals, Type, and Mathematics Extended **Module 3 Lab: Conditionals Practice** Module 3 Project
4	Nesting and Loops	4.1: Nested Conditionals 4.2: Escape Sequences 4.3: 'while' Loops and Incrementing 4.4: 'while' Boolean Loops **Module 4 Lab: Nesting and Loops Practice** Module 4 Project
5	**Unit Assessment**	**Unit 1 Project**

Unit 2 (Five Weeks)

Overview

- Module 1: Sequence Indexes
- Module 2: Sequence Manipulation
- Module 3: Sequence Iteration
- Module 4: Working with Files
- Module 5: Final Evaluation

Objectives

- Access string and substring content
- Iterate through strings
- Create, access, modify, and delete lists and list items.
- Create numeric iteration.
- Extend, sort, and reverse lists.
- Convert between strings and lists.
- Split and join lists
- Import, open, and navigate through files.
- Read and write to files.
- Complete coding assignments related to the content of each module

Module #	Modules	Sections
1	Sequence Indexes	1.1: String Sequences 1.2: Index Slicing 1.3: Iterating Strings 1.4: String Methods **Module 1 Lab: Sequence Index Practice** Module 1 Coding Assignment
2	Sequence Manipulation	2.1: List Sequences 2.2: List Append 2.3: List Insert 2.4: List Delete **Module 2 Lab: Sequence Manipulation Practice** Module 2 Coding Assignment
3	Sequence Iteration	3.1: Power of List Iteration 3.2: Range Iteration 3.3: .extend(), .reverse(), and .sort() Methods 3.4: Between Strings and Lists **Module 3 Lab: Sequence Iteration Practice** Module 3 Coding Assignment
4	Working with Files	4.1: File import, open, and read 4.2: File .readlines() and .close() Methods 4.3: File .readline() and .strip() Methods 4.4: File .write() and .seek() Methods **Module 4 Lab: Working with Files Practice** Module 4 Coding Assignment
5	**Final Evaluation**	**Final Coding Assignment**

Unit 3 (Five Weeks)

Overview

- Module 1: Python Modules
- Module 2: More-Powerful Statements
- Module 3: Methods and Structures for Robust Code
- Module 4: Proper Functions
- Module 5: Final Evaluation

Objectives

- Implement Python modules for system navigation, math, and date-time calculations
- Apply more-powerful statements using compound conditionals (and/or), identity (is), and negation
- Apply complex operator precedence
- Precisely format print output
- Manage errors as a natural part of running code
- Solve problems requiring data collection and recall by using Tuples & Dictionary Data Structures
- Build maintainable and distributable functions and files
- Use standard documentation practices

Module #	Modules	Sections
1	Python Modules	1.1 Using Python Modules 1.2 Working with Dates and Times 1.3 Date and Time Arithmetic 1.4 File System **Module 1 Lab: Python Module Practice**
2	More-Powerful Statements	2.1 Boolean Expressions and Compound Conditionals 2.2 Advanced Loop Structures 2.3 Containment, Identity, and Operator Precedence 2.4 Powerful Output Formatting **Module 2 Lab: Statement Practice**
3	Methods and Structures for Robust Code	3.1 Error Handling 3.2 Files 3.3 Tuples 3.4 Dictionaries **Module 3 Lab: Method and Structure Practice**
4	Proper Functions	4.1 Script Environment & Command-Line Arguments 4.2 Variable Scope 4.3 Documenting Functions (docstrings) 4.4 Documenting Functions (pydoc) Module 4 Lab: Function Practice
5	**Final Evaluation**	**Final Coding Assignment**