



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

Nhân bản – Phụng sự – Khai phóng

# Data Structures & Algorithms

Course Introduction

- **Course Information**
- **Objectives**
- **Contents**
- **Assignments and Labs**
- **Grading Information**
- **Textbook and References**

- **Prerequisite**
  - Programming in C/C++
- **Number of credits: 03**
  - Theory sessions: 02 credits
  - Practice sessions: 01 credits
- **Plan for 15 weeks**
  - 2 theory periods / session / week
  - 2 practice periods / session / week

- Understand and present data structures and algorithms
- Understand graph representations and algorithms
- Understand algorithm analysis
- Be able to translate high-level, abstract data structure and algorithm descriptions into concrete code
- Be able to apply data structures and algorithms to solve problems
- Develop communication, thinking and problem solving skills
- Develop professional attitude

- Course introduction
- C/C++ Review
- Algorithm Analysis
- Linked Lists
- Stacks
- Queues
- Searching & Sorting Algorithms
- Binary Tree
- Binary Search Tree
- Heaps
- Graphs
- Hash Table

- **Assignments**

- Assignments are given after theory session, students should do in class and at home

- **Labs**

- 8 labs will be done
- Solving some problems by using data structures and algorithms

- **Participation, Attendance: 10%**
- **Assignments, Labs: 20%**
- **Midterm exam: 20%**
  - Written test
- **Final exam: 50%**
  - Multiple choice questions

## • Textbook

- M.A. Weiss, Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson, 1997

## • References

- Jiman Hong, Nguyễn Văn Lợi, Nguyễn Phương Tâm, Ninh Khánh Chi, Cấu trúc dữ liệu, Nhà xuất bản TT&TT, 2019
- Đỗ Xuân Lôi, Cấu trúc dữ liệu và giải thuật, ĐHQG Hà Nội, 2006



- Data structures: conceptual and concrete ways to organize data for **efficient storage** and **efficient manipulation**
- Employment of this data structures in the design of **efficient algorithms**
- **Analysis** of the algorithm complexity

- **Requirements for a good software**

- Good detailed design
- Easy maintenance
- Reliable
- Easy to use
- Reusable

⇒ **Efficient data structure**

⇒ **Efficient algorithms**

- **We want to develop a program to manage students including some functions:**
    - Adding new students
    - Modifying student information
    - Searching and sorting students
    - Storing student information
- ⇒ **How can you solve this problem?**



**Nhân bản — Phụng sự — Khai phóng**



**Enjoy the Course...!**