

# **Data Structures and Algorithm with Python (CSD203)**

**Enroll Key:  
CSD203\_HOADNT**

**hoadnt@fe.edu.vn**

# Prerequisites

- **Completed:** PFP191

# Course Objectives

- The course provides an **introduction to data structures and algorithms, including their design, analysis, and implementation.**
- In the main portion of the course, it presents **fundamental data structures and algorithms** such as: **List, Stacks, Queues, Trees, Hash Tables, graph and Text Processing.**

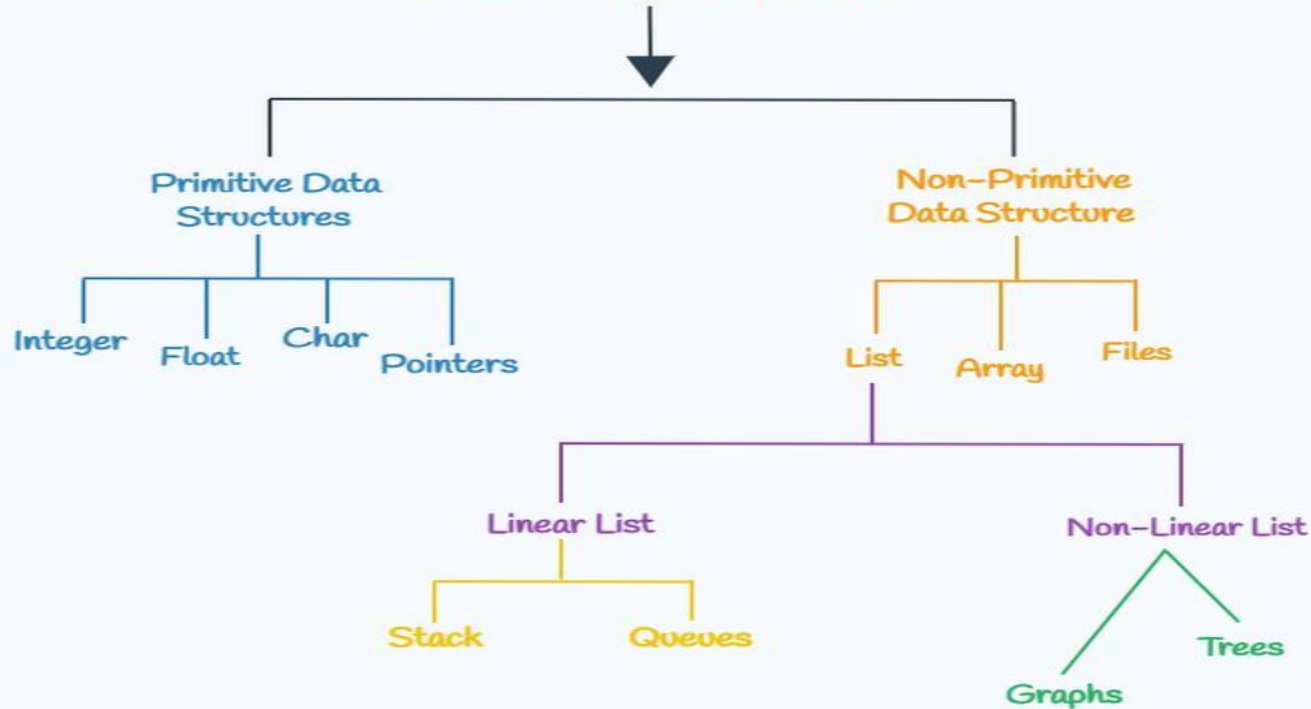
# Course Objectives



- How do you find a book from a library?
  - Method #1: You can check each book one by one until you find the wanted
  - Method #2: You can first locate the bookshelf according to the category of a book, whether it is humanity, or science, or computer science, and then you search in the specific bookshelf.

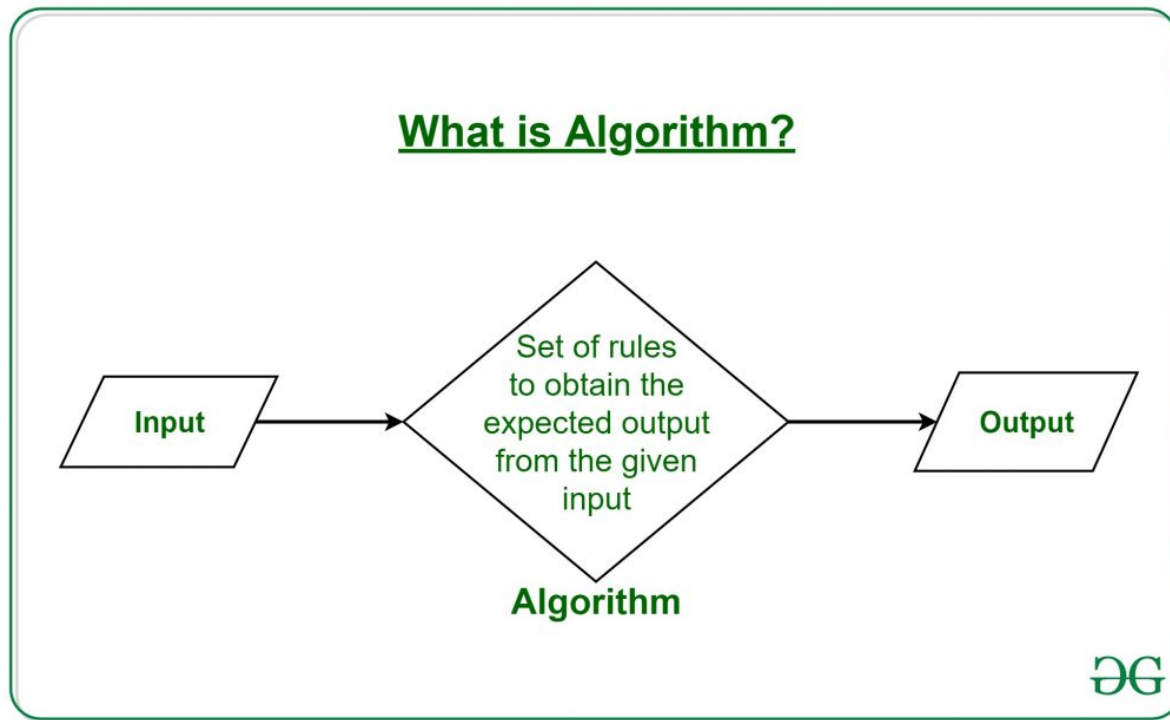
# Course Objectives

## TYPES OF Data Structures



➔ The data structure is a storage structure for a set of data.

# Course Objectives



➔ Algorithm is the methods for solving problems, which could be implemented in programming.

# Course Objectives

- **Data Structure** studies the organization of data in computer:
  - Goal: organize data
  - Criteria: facilitate efficient
    - Storage of data
    - Retrieval of data
    - Manipulation of data
- **Algorithm**
  - Outline, the essence of computational procedure, step by step instruction.
- **Program = data structures + algorithm**

# Course Plan

1. Describe the list data structure and its' different way of implementation. Implement the singly linked list.
2. Define stack and queue. Describe basic operations and the use of these structures.
3. Describe recursive definitions, algorithms, functions, and their implementation and use.
4. Explain the general tree, Binary Tree, and Binary Search Tree (BST). Implement BST with basic operations.



# Course Plan

1. Discuss about graphs and their application. Implement a graph with some basic operations.
2. Explain the operation and performance of some basic and advanced sorting algorithms.
3. Explain hashing and application.
4. Describe the Text Processing problem and its' application. Explain the Huffman, LZW, and Run-length encoding Algorithms.

# Materials/ References

- **Required Textbook**
  - Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser
- **Required References**
  - Data Structures and Algorithms in Python, Rance D. Necaise
  - Data Structures and Algorithms with Python, Kent D. Lee  
Steve Hubbard
- **References**
  - Google ☺

# Learning Environments

- Python 3x: <https://www.python.org/>
- Pycharm: <https://www.jetbrains.com/pycharm/>

# Course Rules

- **How to conduct**
  - Prepare contents of the next session/ topic at home
  - Following lessons in classrooms
  - Completing chapter assessments in time and Quizzes (via LMS)
- **Communication**
  - Class
  - Interchange by FU-HCM CMS, Forum
  - Discussing actively in your teams and in classrooms
  - Free to question and answer
- **Others**
  - Off phone
  - Use laptops under teacher's instruction
  - No game, no chat in class

# Evaluation Strategy

- Must attend more than 80% of contact hours (if not, not allow to take exam).
- **Evaluating**
  - **02 Progress Test (Q)** **20 %**
    - Quiz 1
    - Quiz 2
  - **N Workshop+ assignment** **20 %**
    - Slide: labs
    - Workshops
    - Assignments
  - **01 Practical Exam (PE)** **30 %**
    - 1 slot.
  - **Final Exam (FE)** **30 %**
- **Total score**
  - 20% (Qiz) + 20% (Assignment) + 30% (PE) + 30% (FE)
- **Pass**
  - Total score  $\geq 5$  and Final Examination  $\geq 4$  (of 10)
  - Every components  $> 0$
  - PE  $> 0$



# How to study

- This course is **complex knowledge** (however, it's **attractive and exciting**), so you need to keep tight grip on it
  - **Read**
    - On the books to get the general concept
    - Reference, study, collection from anywhere else (internet, your classmates, forum ...)
  - **Attend lectures**
    - Listens, understand, then make your own notes
    - Give your explanation about some topic in lectures
    - Ask questions
    - Give some examples that are not existed in your book
    - Practice all the exercises, demo to make your sense
  - **After classes**
    - Discuss your classmate in directly, on forum
    - Do the lab, assignments to submit via CMS, and do more exercises
    - Build your teams in yourselves to support together in studying

# How to exam/test

- This course is **required** following rules, so you **need to focus and practice** your exercises and homework in try your best everyday
  - **Progress Tests**
    - **No** books
    - **No** conversations
  - **Practical Exam**
    - **No** internets, **No** emails, **No** chats, **No** conversation
    - **Not copy or paste** from available/previous code.
    - All are try it yourselves **manually**
    - ... Nothing else
    - You do only work with Pycharm **IDE** tools
  - **Workshop/Assignment**
    - **Not** copy (copy code, contents, style)
    - Submission of all source code **does not delete anything**

# Academic policy

- Cheating, plagiarism and breach of copyright are serious offenses under this Policy.
  - Cheating
    - Cheating during a test or exam is construed as talking, peeking at another student's paper or any other clandestine method of transmitting information.
    - Cheating during in making lab and assignment as copy source code, copy style, same meaning in progress, ...
  - Plagiarism
    - Plagiarism is using the work of others without citing it; that is, holding the work of others out as your own work.
  - Breach of Copyright
    - If you photocopy a textbook without the copyright holder's permission, you violate copyright law.



# Enjoy the Course

- Be enthusiastic about the material because it is interesting, useful and an important part of your training as a software engineer.
- Our job is to help you learn and enjoy the experience.
- We will do our best but we need your help.
- So, let's all have fun together with CSD203!!!

Hoadnt@fe.edu.vn

Q & A