Course Introduction

DASC 5300: Foundations of Computing

Alex Dillhoff

The University of Texas at Arlington

Course Overview

Course Description

Basics of programming, data structures, and algorithms. Introduction to databases and operating systems. Basics of discrete structures and computability. Course is used for the Master's in Data Science degree program and certificate programs for non-CSE majors.

What will you learn?

- **Python**: Extremely popular language for data science
- **Data Structures and Algorithms**: How to represent data and manipulate it
- **Computational Complexity**: How to measure the efficiency of algorithms
- **Databases**: Big data cannot fit in memory
- **Data Analysis**: How to clean and analyze data

Programming with Python

What is Python?

- Python is a **high-level**, **interpreted**, **object-oriented** programming language.
- Python is **dynamically typed** and **garbage-collected**.
- Python is **multi-paradigm**: it supports procedural, object-oriented, and functional programming.
- Python is **cross-platform**: it runs on Windows, Mac, Linux, and many other platforms.

Why Python?

- Python is **easy to learn** and **easy to read**.
- Python is widely used in **data science**, **machine learning**, and **artificial intelligence**.
- There are many data science-related libraries for Python, such as **NumPy**, **Pandas**, **Matplotlib**, **Scikit-learn**, and **PyTorch**.

Expectations

- You won't master Python in one semester.
- Programming is a skill that requires practice.
- You will learn the basics of Python along with foundational applications in computer science.

Data Structures and Algorithms

Data Structures

- A **data structure** is a way of organizing data in a computer so that it can be used effectively.
- We will cover **arrays**, **linked lists**, **stacks**, **queues**, **trees**, **graphs**, and **hash tables**.
- We will also look at **tensors** (multi-dimensional arrays) and how they are handled them in practice.

Algorithms

- An **algorithm** is a set of instructions for solving a problem.
- We will cover various algorithms for sorting, searching, and graph traversal.
- Each algorithm is commonly tied to a particular data structure.

Computational Complexity

Measure the efficiency of algorithms

- **Computational complexity** is a measure of the amount of resources required to run an algorithm.
- We will cover **Big-O notation** and how to use it to analyze the efficiency of algorithms.
- Complexity is measured in both **time** and **space**.
- Time complexity relates to the algorithms we use, and space complexity relates to the data structures.

Databases

Storing Big Data

- **Big data** is data that is too large to fit in memory.
- Most of the algorithms we will cover up to this point assume that all data fits in memory.
- We will cover **relational databases** and **SQL**.
- We will also look at disk-based versus in-memory databases.

Data Analysis

Working with raw data

- The last part of the course will focus on introductory **data analysis**.
- Raw data is often messy and needs to be cleaned before it can be analyzed.
- We will cover **data cleaning**, **data visualization**, and **data analysis**.
- We will use **Pandas**, **NumPy**, and **Matplotlib**.

Course Structure

Course Structure

- The course is divided into 3 primary modules.
- Each module will have at least 1 programming assignment.
- There will be a quiz roughly every 2 weeks.
- There are 3 exams (including the final).

How to be successful

- **Practice** programming often.
- **Start early** on assignments.
- **Ask questions** when you need help.
- **Review** material before and after class.

Questions?