Definitions

**Modularity** is the division of a program into independent, reusable pieces.

**Data hiding** is the protection of data from other parts of the program.

An **abstract data type (ADT)** specifies a set of data items and operations on those items. An ADT does not include details of how items are stored or how the operations are implemented.

A **data type** is a data storage format.

A **data structure** is one particular way of storing data items, with each operation implemented by an **algorithm**.

For now, **pseudocode** is a way of describing an algorithm without specifying a particular machine or programming language.
Viewpoints of user/provider

1. Determine data types and operations.
2. For each type, choose an ADT.
3. Develop pseudocode algorithm using ADT operations.
4. Calculate cost of algorithm with respect to cost of operations.
5. Receive information on cost of data structures.
6. For each ADT, choose the best data structure for the numbers of operations used.

1. Create pseudocode of implementations of the ADT using various data structures, using the pseudocode interface.
2. Analyze data structures.

Establish the code interface.
Code the algorithm using the interface.

Request to perform operation
Code the chosen data structure using the code interface.

Result of operation
Recipes for planning

Solving a problem (user/plan):
1. Determine data types and operations.
2. For each type, choose an ADT.
3. Develop pseudocode algorithm using ADT operations.
4. Calculate cost of algorithm with respect to costs of operations.
5. Receive information on cost of data structures.
6. For each ADT, choose the best data structure for the numbers of operations used.

Choosing among data structures (provider/plan):
1. Create pseudocode of implementations of the ADT using various data structures, using the pseudocode interface.
2. Analyze data structures.
Preconditions and postconditions

A **precondition** is a requirement that must be satisfied for an operation to be guaranteed to work.

A **postcondition** is a guarantee of the outcome of an operation being executed.
Recipes for coding

User/code:
- Establish preconditions and postconditions in the code interface.
- Code the algorithm using the code interface.

Provider/code:
- Code the chosen data structure using the code interface.
Course logistics

Components:

- Lectures
- Readings (textbook optional, added resources are not)
- Python self-study
- Self-checks (only in class)
- Assignments
- Exams

Tools used:

- Course website: schedule, resources, news; lecture summaries in advance and some updates after
- Piazza: complete survey; discussion
- MarkUs: hand in assignments; check marks
Work to do this week

- Start on Python review (see Python page on website, and schedule).
- Attend a Python review session, if possible, or use the resources on the Python page.
- Read Python 2 versus Python 3 page if you studied Python a while ago.
- Sign up for Piazza.
- Complete survey by Monday (on Piazza).