



# Data Structures and algorithms

ABSTRACT DATA TYPES (ADTS)

WEEK 1

LECTURER: DR ANWARUL PATWARY

# Outline

- ▶ What is an abstract data types?
- ▶ What do use them for?



# Abstract Data Types

- ▶ It's another data structure
- ▶ It's a collection of data whose representation is hidden.
- ▶ It only defines the operations for manipulating data but not how that structure is implemented.
- ▶ E.g. stacks, queues.
- ▶ Stack can be done with an array OR with a linked list.

# Operations

- ▶ If we define a data structure in terms of what we can do with it –rather than in terms of exactly how it is arranged in memory – then we have specified the operations we can perform on it.
- ▶ Such a specification is an Abstract Data Type.



# Object-oriented languages

- ▶ Object-oriented programming is based around the concept of abstract data types.
- ▶ Java is an object-oriented programming language and java classes are ideal for implementing ADTs.

# What is needed to make an ADT?

- ▶ Variables for holding data.
  - ▶ Usually hidden from the user.
- ▶ Operations that can be performed on the data.
  - ▶ These must be available to the user.



# Information hiding

- ▶ ADTs support information hiding.
- ▶ Variables can be made private, meaning there is no access to the variables by users of class.
- ▶ Method can be made public, so users of the class can create and manipulate the data structure.

# Modularity

- ▶ ADTs provide *modularity*. What does this mean?
- ▶ It means that –as long as the public parts of the ADT have been clearly specified – other parts of our program can be developed, maintained or upgraded independently of our ADT.
- ▶ They don't depend on *internal details* of the ADT, so we can even start writing them before the ADT is done.
- ▶ Or, if we change our mind about how to implement the ADT – as long as the public interface remains the same, this won't affect other parts of our software.



# Other benefits

- ▶ Polymorphism: the same operation can be applied to different types.
- ▶ Inheritance: where subclass adopt from parent classes.