

Data Structures

- A **data structure** is a specialized format for organizing, processing, retrieving, and storing data.
 - An **abstract data type** is the “big picture” of a data structure:
 - it contains the collection of data,
 - and the collection of functions on the data.
1. An array is an abstract data type:
 - An Array contains a sequence of items.
 - **create** (A, n) creates an array A for storing n items.
 - **set** ($A, i, item$) stores $item$ in the i^{th} position in the array A .
 - **get** (A, i) returns the value of the item stored at the i^{th} position in the array A .

Class and Object-Oriented Programming

- To use an abstract data type, we can implement it into a **class**.
 - An abstract data type is still a data type, just like other data types such as integer, char, Boolean... We handle it just like a type of object.
 - A class is a template definition of the methods and variables in a particular kind of object.
2. To implement the abstract data type Array into a class:
 - **create** (A, n) can be implemented as $A.items = [] * n$
 - **set** ($A, i, item$) can be implemented by $A.items[i] = item$
 - **get** (A, i) can be implemented by return $A.items[i]$
- The using of abstract data types and classes is a feature of **Object-Oriented programming**. Our programming is relied on objects.
 - For example, if we want to pop out the top item from a stack S , Object-Oriented programming might give the following function: $S.pop()$; but Functional-Oriented programming might give the following function: $pop(S)$.

Algorithms

- An **algorithm** is a finite sequence of precise instructions for performing a computation or for solving a problem.
 - To describe an algorithm, we can use **pseudo-code**. A pseudo code is something between human languages and programming languages.
3. How to find the maximum number in an array?

```

max ( $A[p \dots r]$ )
1  $max = A[p]$ 
2   for  $i = p + 1$  to  $r$ 
3     if  $max < A[i]$  then  $max = A[i]$ 
4 return  $max$ 

```

- This is not any real programming language, but if you understand any programming language you can easily understand it.
- Sometimes, a pseudo-code can be really human-language-like. For example, in a completed algorithm, the above procedure might be expressed with only one line:

let n be the max in $A[p \dots r]$

As long as it is clear, an instruction in a pseudo-code can be written in any way you like.

- In this course, we also care about the realization/implementation of pseudo-code. To implement the above pseudo-code in python, we have:

```
def max (array: [int]) -> int:
    m = array[0]
    for i in range (1, len(array)):
        if m < array[i]:
            m = array[i]
    return m
```

Atomic Data Types in Python

- An **atomic data type** is a single, inseparable unit of data. There are four types of atomic data types in Python: integers, floats, characters, and Booleans.

- Integers and floats are used to handle numbers.

4. What are the results of the following operations?

- $2+2.34$
- $2+3*4$
- $(2+3)*4$
- $2**10$
- $6/3$
- $7/3$
- $7//3$
- $a, b = 5, 4$
while $a > 0$:
 $b = b - 1$
 $b = b - 1$
 $a = a - 1$
print(b)

- Booleans are used to handle truth values.
- Characters can be expressed with either `" "` or `' '`.

5. What are the results of the following operations?

- True and False
- $4.4 \leq 4$
- `"A" < 'a'`
- $1 == '1'$
- $2*2 != 4.00$
- `"a" + "a"`

- "a" + "1"
- "1" + 1
- 'a' * 3
- True + False
- True + 2
- False == 0

List

- **List** is a type of built-in abstract data type in python. Since python doesn't have array class, arrays are handled as Lists in python.
- A list can contain zero or more items, they are ordered, and they don't have to have the same data type.

6. Try the following operations in the list below

```
mylist = [1.2, 5, True, "baby"]  
print(mylist)  
print(len(mylist))  
print(mylist * 3)  
print(mylist[1:3])  
print(mylist[1:4])  
print(mylist[1:20])
```