Al-Mustaqbal University
College of Healthcare and Medical Techniques
Intelligent Medical Systems Department

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية التقنيات الطبية والصحية

قسم الانظمة الطبية الذكية

Subject: Data Structure

Class: Second

Lecturer: Asst. Prof. Mehdi Ebady Manaa

Lecture: (2)

Overview of Algorithms

Overview of Algorithms

Many of the algorithms we'll discuss apply directly to specific data structures. For most data structures, you need to know how to:

- Insert a new data item.
- Search for a specified item.
- Delete a specified item.

The concept of **recursion** is important in designing certain algorithms. Recursion involves a method **calling itself**.

Problems with Procedural Languages

OOP was invented because procedural languages, such as C, Pascal, and early versions of BASIC, were found to be inadequate for large and complex programs. **Why was this?**

There **were two kinds of problems**. **One** was the lack of correspondence between the program and the real world, **and the other** was the internal organization of the program.

1. Poor Modeling of the Real World

Conceptualizing a real-world problem using procedural languages is difficult. Methods carry out a task, while data stores information, but most real-world objects do both of these things.

For large programs, which might contain hundreds of entities like thermostats, this procedural approach made things chaotic, error-prone, and sometimes impossible to implement at all. What was needed was a better match between things in the program and things in the outside world.

2. Crude Organizational Units

A more subtle, but related, problem had to do with a program's internal organization. Procedural programs were organized by dividing the code into methods. One difficulty with this kind of method-based organization was that it focused on methods at the expense of data. There weren't many options when it came to data.

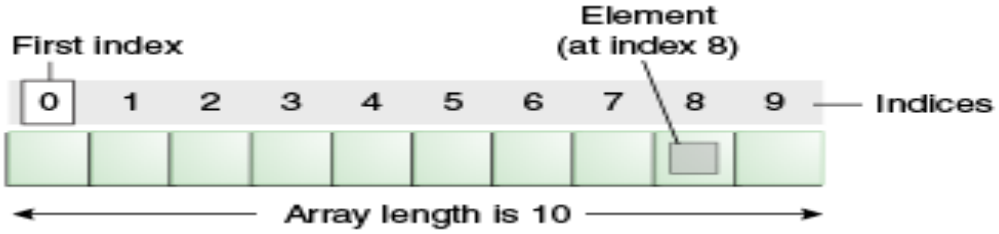
To simplify slightly, data could be local to a particular method, or it could be global—accessible to all methods. There was no way (at least not a flexible way) to specify that some methods could access a variable and others couldn't.

This **inflexibility** caused problems when several methods needed to access the same data. To be available to more than one method, such variables needed to be global, but global data could be accessed inadvertently by any method in the program. This lead to frequent programming errors. What was needed was a way to fine-tune data accessibility, allowing data to be available to methods with a need to access it, but hiding it from other methods.

Arrays

The array is the most commonly used data storage structure; it's built into most programming languages.

We can define an array, as numbered collection of variables all of the same type. Each variable, or cell, in an array has an index, which uniquely refers to the value stored in that cell. The cells of an array a are numbered 0, 1,2, and so on.



The above diagram illustrates that:

- An array is a container of elements
- Elements have a specific value and data type, like integer, floats, etc.
- Each element also has its own index, which is used to access the element.

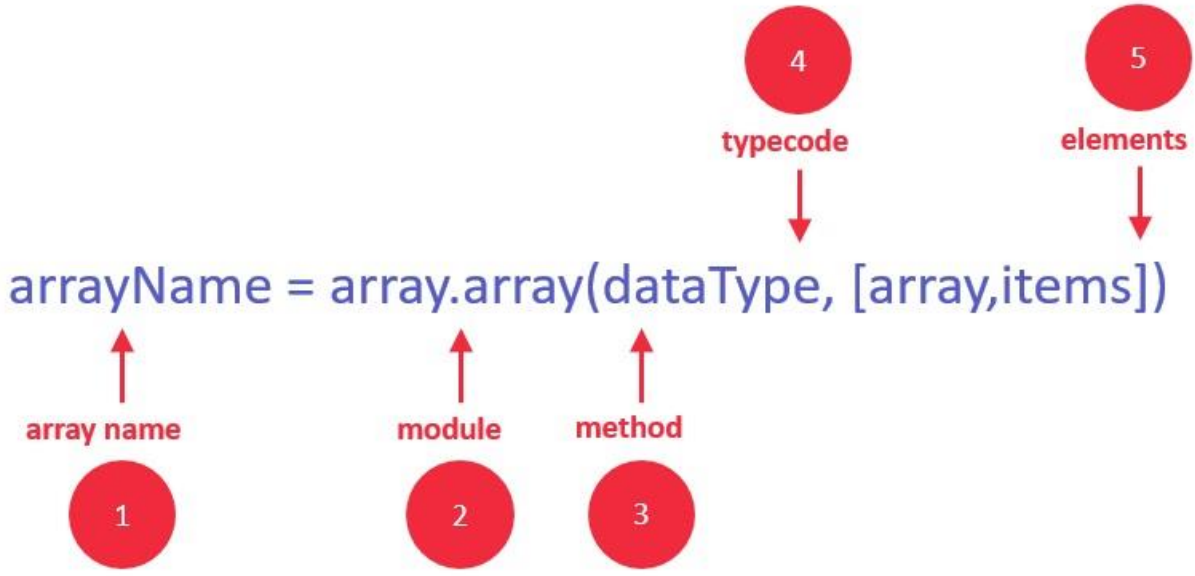
Notes:

- Elements are stored at contiguous memory locations.
- An index is always less than the total number of array items.
- In terms of syntax, any variable that is declared as an array can store multiple values.
- However, three parts will always remain common in all the initializations, i.e., array name, elements, and the data type of elements.

Declaring Array Variables

You can declare an array in Python while initializing it using the following syntax.

arrayName = array.array(type code for data type, [array,items])



1. **Identifier**: specify a name like usually, you do for variables

2. **Module**: Python has a special module for creating array in Python, called "array" – you must import it before using it
3. **Method**: the array module has a method for initializing the array. It takes two arguments, type code, and elements.
4. **Type Code**: specify the data type using the type codes available (see list below)
5. **Elements**: specify the array elements within the square brackets, for example [130,450,103]

How to create arrays in Python?

In Python, we use following syntax to create arrays:

```
Class array.array(type code[,initializer])
```

For Example

```
import array as myarray

abc = myarray.array('d', [2.5, 4.9, 6.7])
```

The above code creates an array having integer type. The letter 'd' is a type code.

Following tables show the type codes:

Type code	Python type	C Type	Min size(bytes)
'u'	Unicode character	Py_UNICODE	2
'b'	Int	Signed char	1
'B'	Int	Unsigned char	1
'h'	Int	Signed short	2
'l'	Int	Signed long	4
'L'	Int	Unsigned long	4
'q'	Int	Signed long long	8
'Q'	Int	Unsigned long long	8
'H'	Int	Unsigned short	2
'f'	Float	Float	4
'd'	Float	Double	8
'i'	Int	Signed int	2
'I'	Int	Unsigned int	2

Exercise

Write python program to insert ten item in array

MCQ Questions

Question 1: What are the essential operations for most data structures that need to be performed on a data item?
A. Insert, Update, Remove
B. Insert, Search, Delete
C. Add, Modify, Delete
D. Create, Read, Update, Delete
Answer: B. Insert, Search, Delete
Question 2: Why were procedural languages found inadequate for large and complex programs?
A. Lack of correspondence between program and real world, and limited internal organization

B. Lack of recursion support, and inefficient memory usage
C. Limited syntax, and lack of built-in functions
D. Lack of object-oriented features, and poor error handling
Answer: A. Lack of correspondence between program and real world, and limited internal organization
Question 3: What is the main problem with the method-based organization in procedural programs? A. It focuses too much on data and not enough on methods
B. It allows flexible data accessibility, causing programming errors
C. It focuses on methods at the expense of data, leading to inflexibility
D. It doesn't support recursion, limiting algorithm design possibilities
Answer: C. It focuses on methods at the expense of data, leading to inflexibility
Question 4: In the context of arrays, what does the index uniquely refer to? A. Address in memory where the element is stored
B. Data type of the element
C. Value stored in the array cell
D. Position of the element within the array
Answer: C. Value stored in the array cell
Question 5: What is necessary to declare an array in Python? A. Identifier, Module, Method, Type Code
B. Identifier, Module, Type Code, Elements
C. Identifier, Type Code, Elements
D. Module, Method, Type Code, Elements
Answer: B. Identifier, Module, Type Code, Elements