A-Level Computer Science P4 Notes (Visual Basic)

First Edition

Navid Saqib

(0333-4259883)

Visiting Teacher

Lahore Grammar School

Beacon House School

SISA

KIMS

ROOTS

Title	A-Level Computer Science P4 Notes (Visual Basic)
Author	Navid Saqib. (0333-4259883)
Published by	MS Books (042-35774780)
Edition	First Edition
Legal Advisor	Ashir Najeeb Khan (Advocate High Court) **AKBAR LAW CHAMBERS** 39-40, 1st Floor, Sadiq Plaza, The Mall, Lahore 042-36314839, 0307-4299886
For Complaints/Order	**MS Books** 83-B Ghalib Market, Gulberg III Lahore (042-35774780),(03334504507),(03334548651)

CONTENT TABLE

CHAPTER # 1

COMPUTATIONAL THINKING

Computational Thinking

Computational thinking (CT) involves a set of problem-solving skills and techniques that software engineers use to write programs that underlie the computer applications you use such as search, email, and maps.
Computational thinking is comprises of following elements

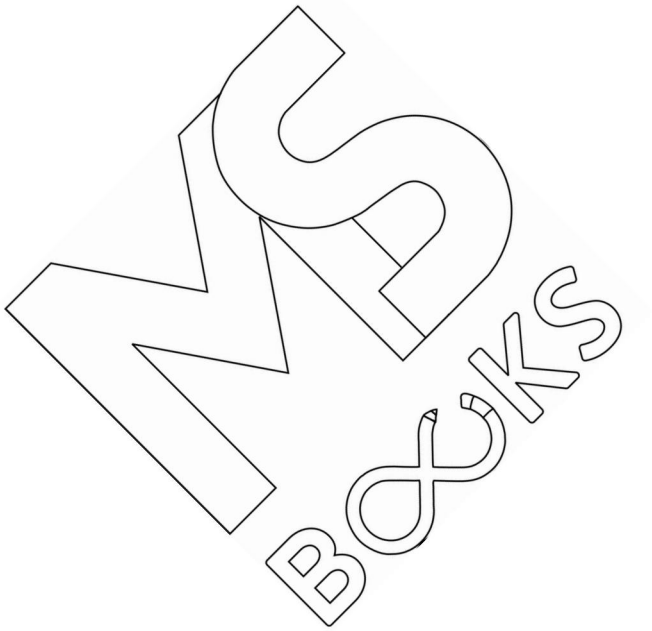
- **Abstraction**
- **Decomposition**
- **Data Modeling**
- **Pattern recognition**
- **Algorithm Design**

Abstraction:

Abstraction means filtering out that information which is not required to solve the problem. E.g. out of complete school records dropping all others student record to obtain the data of A 2 Students who Takes computer as subject

Or a filtered detail about a particular object e.g. filtering out a metro station map out of complete map is abstraction. Because metro administration want to show all the stations of the route in the map, complete city map is not required

Note: Abstract for the computer scientist is intended to hide unwanted detail and so to simplify our understanding.



Decomposition

Decomposition is breaking down the tasks to smaller tasks in order to explain or understand the task. Decomposition is another word to step wise refinement, Also known as top down design technique as well

1. **Setting and declaration**

1.1 Variables
1.2 Consents

2. **Input**

2.1 Get Num1 input
2.2 Get num2 input

3. **Process**

3.1 Get total by adding num1 and num2

4. **Output**

4.1 Output user message
4.2 Output total

Data modeling:

Data modeling involves analyzing and managing the data using and defining data types like (integer, String, Date, double, single, Boolean known as composite data types).

1D arrays to represent Lists
2 D Arrays to represent Tables
Handling data model in Files (text Files)

Will Learn **Abstract Data Types** which will be part of Data Modeling

Pattern recognition:

Means looking for common patterns or COMMON SOLUTIONS TO THE COMMON PROBLEMS Using them to complete the task in more effective and efficient way, Standard algorithm like Bubble Sort, Insertion sort and binary search are the part of pattern recognition

Algorithm Design:
Algorithm design is following the step by step approach to solve the problem, also known as the Basic pattern of designing the algorithm.

Standard Patterns:

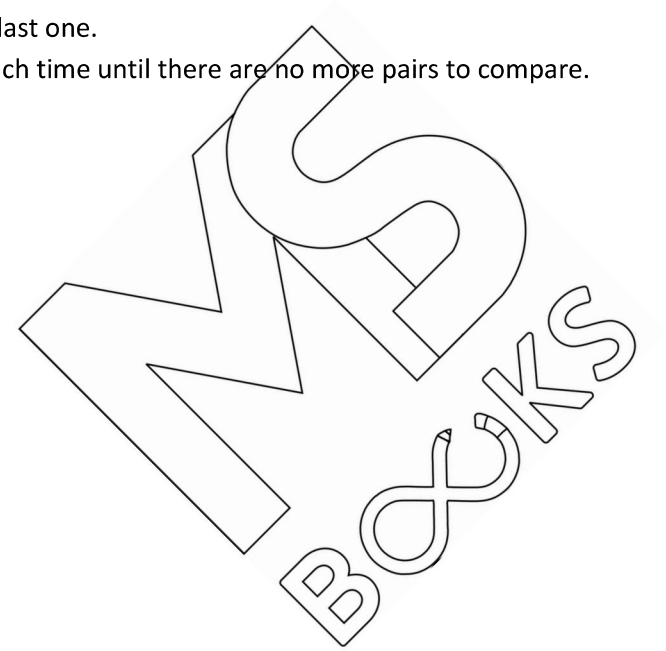
- **Bubble Sort**
- **Insertion sort**
- **binary search**

Bubble sort

Comparing each pair of values in an array and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. Bubble sort is pattern which use array as a main source and compares two items at a time. It SWAPS these two items if they are in the wrong order. In an attempt when it compare or check all values in array that is known as PASS. It continues to check the array until no swaps are needed that means the array is sorted. The algorithm gets its name from the way larger elements "bubble" to the top of the list. It is a very slow way of sorting data and rarely used in industry. There are much faster sorting algorithms out there such as insertion sort and quick sort

How it works:

1. Compare adjacent values of array. If the first is greater than the second, swap them.
2. Repeat this for each pair of adjacent Values, starting with the first two and ending with last two. At this point the last element should be the greatest.
3. Repeat the steps for all elements except the last one.
4. Keep the repeating for one fewer element each time until there are no more pairs to compare.



Let’s check this code

Module1.vb*

Module1

```
Module Module1

    Sub Main()
        Dim Valueslist(4) As Integer
        Dim Temp As Integer
        Dim starting, ending As Integer
        starting = LBound(Valueslist)
        ending = UBound(Valueslist)
        'array input
        For x = starting To ending  'can be 0 to 4
            Console.WriteLine("Enter Values in Array")
            Valueslist(x) = Console.ReadLine
        Next
        'display output to user'
        Console.WriteLine("Values in Array")
        Console.WriteLine("")
        For y = starting To ending  'can be 0 to 4
            Console.WriteLine(Valueslist(y))
        Next
        'Bubble sorting iteration'
        For i = starting To ending
            For K = starting To ending - 1
                ' why -1 because last value cant be compared '
                If Valueslist(K) > Valueslist(K + 1) Then
                    Temp = Valueslist(K)
                    Valueslist(K) = Valueslist(K + 1)
                    Valueslist(K + 1) = Temp
                End If
            Next
        Next
        'display Sorted output to user'
        Console.WriteLine("Sorted Values in Array")
        Console.WriteLine("")
        For z = starting To ending  'can be 0 to 4
            Console.WriteLine(Valueslist(z))
        Next
        Console.ReadKey()
    End Sub

End Module
```

file:///C:/Users/Bunny/AppData/Local/

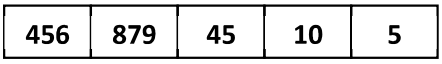
```
Enter Values in Array
456
Enter Values in Array
879
Enter Values in Array
45
Enter Values in Array
10
Enter Values in Array
35
Values in Array

456
879
45
10
35
Sorted Values in Array

10
35
45
456
879
```

How it works?

The initial values stored in the array are as follows:

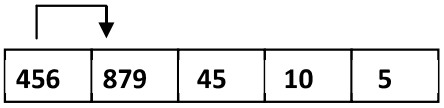


In the above example, nested loops are used to sort the array. The outer loop moves from 0 to 4 which is starting to ending and with each iteration of outer loop, inner loop moves from Starting to ending -1 or we can say 0 to 4 - 1.

First of all, the value of i is 0 so the focus of outer loop is on the first Value of the array. The sorting process will work as follows:

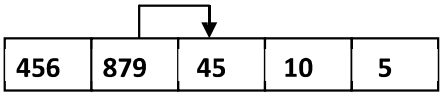
Pass 1 Started

Iteration 1

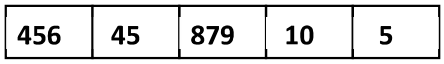


K =0 so the statement if valuelist(k) > Valuelist(K+1) compares 456 with 879 As 456 is not greater than 879, there will be no change in the array.

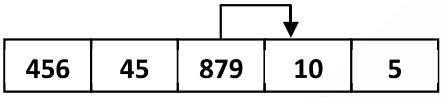
Iteration 2



K=1 so the statement if valuelist(k) > Valuelist(K+1) compares 879 with 45. As 879 is greater than 45, both values will be interchanged and the array will be as follows:



Iteration 3



K=2 so the statement if valuelist(k) > Valuelist(K+1) compares 879 with 10. As 879 is greater than 10, both values will be interchanged and the array will be as follows: