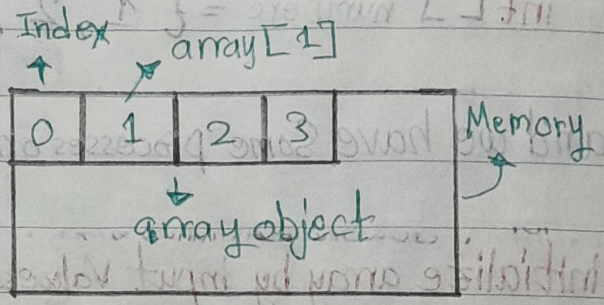
Chapter 7 :-

Array Basics :

Def: is a box make you able to store more than one value in the same variable you can treat it as an instance or object and you can imagine it as following figure :



and we can declare an array as following Syntax :

```
elementType [] arrayRefrenceVar ;
```

and we can create it as following Syntax :

```
elementType [] arrayRefVar = new elementType[arraySize];
```

and we can assign a value as following Syntax :

```
arrayRefvar[i] = value;
        ↑
      index
```

notes The Size of array cannot be changed after the array is Created

notes The array is always ~~based~~ based 0 and if the array length is i then the last index is i-1

we can make initiallization for an array as following Syntax:

```
int[] numbers = { 1, 2, 3, 4 };
```

and we have some processes on arrays:

1. Initialize array by input Values:

```
for (int i=0; i< arr.length ; i++) {
    arr[i] = input.nextInt();
```

2. Displaying array elements:

```
for (int i=0; i< arr.length ; i++)
    System.out.println(arr[i] + " ");
```

3. Summing array elements:

```
for(int i=0; i< arr.length ; i++)
    total += arr[i];
```

[4] Finding the largest element:

```
double max = list[0];
for (int i=0; i< list.length; i++) {
    if (list[i] > max)
        max = list[i];
```

[5] Finding the smallest index of the largest element:

```
double max= list[0];
int indexOfMax = 0;
for (int i =0; i< list.length; i++) {
    if ( max < list[i]) {
        max = list[i];
        indexOfMax = i;
    }
```

[6] Random Shuffling:

```
for ( int i = mylist.length-1; i >0; i--) {
    int j = (int) ( Math.random() * i+1);
    double Temp = mylist[i];
    mylist[i] = mylist[j];
    mylist[j] = Temp;
}
```

[7] Shifting Elements of array:

```
double Temp = mylist[0];
for (int i = 1; i < mylist.length; i++){
    mylist[i-1] = mylist[i];
}
mylist[mylist.length -1 ] = Temp;
```

and you can loop on array with foreach loop as:
The following Syntax:

```
for (elementType element : arrayRefVar) {
    System.out.println(e); -> process
}
```

notes: when you loop on index ~~do~~ which isn't exist you will see the Compiler make an exception called ArrayIndexOutOfBoundsException

---

Copying Arrays:

you can copy any array by System.arraycopy(); method

```
System.arrayCopy(SourceArr, srcPos, TargetArr, tarPos,
                 length);
```

Passing Arrays to Methods:

you can treat the array as an arguments which can be passed as the following Syntax:

```
int[ ] numbers = { 1, 2, 3};
printArray (numbers);
public static void printArray (int[ ] n){
    for (int i : n)
        System.out.print ( i + " ");
}
```

notes When you pass an array you are using pass by Value.

notes you can creat an array by the following Syntax:

```
new elementType[ ] { Value0, ____, ValueK};
```

Passive Array ↷ Anonymous Array

notes The Difference between pass by Value and pass by Sharing is to passing the Value to a method but out side it will not change, but if ~~use~~ you used pass by Sharing The Value will change outside and inside the method by sharing The reference Variable

notes The array stored in the Heap of memory

Returning an array from a Method:

you can Return an array by it's Reference Variable as the following Syntax:

```
public static int[] reverse (int[] list) {
    int[] result = new int[list.length];
    for (int i=0, j=result.length-1;
         i < list.length; i++, j--) {
      result[j] i = list[i];
    }
    return result;
}
```

notes

- you can pass more than one variable had The same Type and the Method will treat them as an array

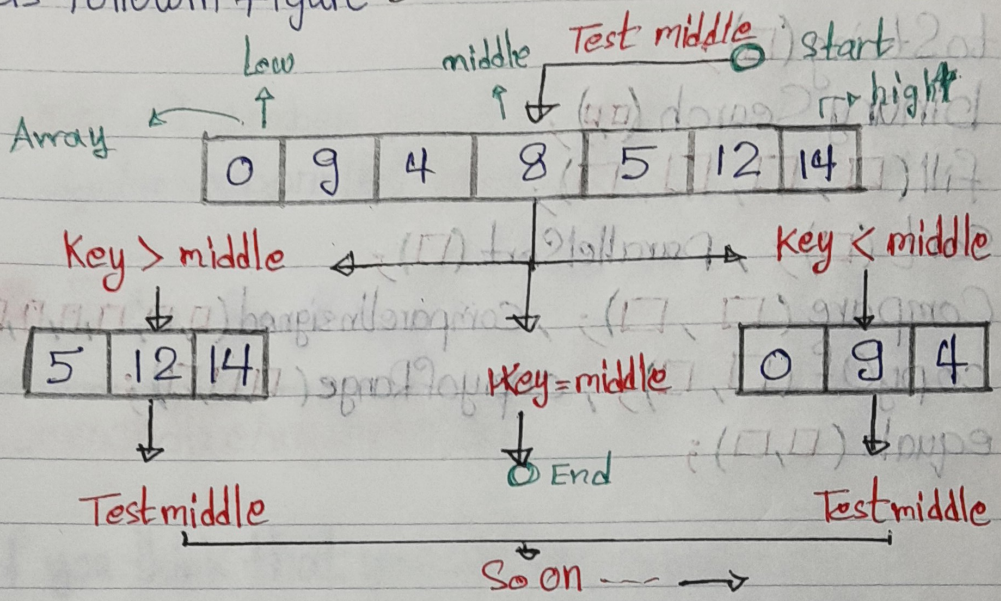
---

Searching Arrays:

you have 2 ways to search in array:

1. LinearSearch ⇒ Comparing The key of elements with each key element in the array.
2. BinarySearch ⇒ is to cancel any position which not be in Searching Process

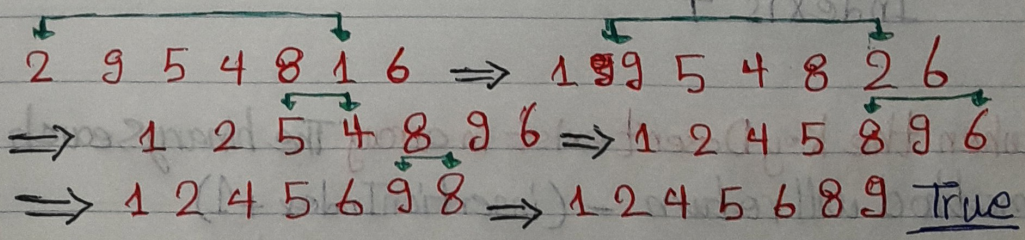
and we can represent the Algorithm of Balnary Search as following figure :



note: Before using Binary Search The array Should be Sorted

Sorting Arrays :

Def: is a Type of Swaping Principle to Sort the elements of array as your wish and we can interpret it as following figure :-



Arrays Class: Methods

1. toString(□);
2. binarySearch(□,□);
3. fill(□,□,□,□);
4. sort(□); , parallelSort(□);
5. Compare(□,□); , CompareUnsigned(□,□,□,□,□,□);
6. copyOf(□,□); , copyofRange(□,□,□);
7. equals(□,□);
8.

binarySearch(array[], key) : is a method which search on the specified key in an array

5 → insertionIndex

ex:
```
int[] list = {2,4,7,10,11,45,50,59};
Arrays.binarySearch(list, 12);
Arrays.binarySearch(list, 11);
```

output:
```
Index is -6
Index is 4
```
→ -(insertionIndex+1)

note: when the key Dosen't exist in array The binarySearch method will return -(insertionIndex+1)

The Index Assumed To be The Key in

toString(array[]); : is a method which turn the whole array into String

ex:

```
int[] numbers = {1, 2, 3}
System.out.println(Arrays.toString(numbers));
```

output: [1, 2, 3]

CommandlineArguments:

If you think that you can pass arguments for The main method you are true. The main method can Take arguments Throgh The args[] you can use The following Command:

```
java filename/classname  arg0  arg1  arg2
```

you can use strings in main without Qoutes Mark
