

## ⇒ Java Objects

- A java object is an instance of a class
- How an object be created. (using new keyword)
- object encapsulate both data (fields) and behaviour (method) into a single unit.

## ⇒ Object contains :-

- State :- it means attribute / fields of object
- Behaviour :- it means method of object.
- Identity :- it means a unique name of obj.

Note :-

When we create an object which is non-primitive data type, its always allocated on the heap memory.

Syntax:-

ClassName objName = new ClassName();

Ex:- Car myCar = new Car();

---

---

## ⇒ Initialise an object

- 1) By reference variable
- 2) By method
- 3) By constructor

### → By reference variable :-

```
public class Main {  
    public static void main(String[] args) {  
        // initialise an object using reference variable  
        Main myObj = new Main();  
  
        myObj.myMethod();  
    }  
  
    public void myMethod() {  
        System.out.println("Object initialised successfully");  
    }  
}
```

object is declared and  
initialised in a single line  
using new keyword

# → Anonymous Object

an object which doesn't have any name

it cannot be used anytime in future

It is a single line operation.

```
public class Main {  
    public static void main(String[] args) {  
        // initialise an object using reference variable  
        new Main().myMethod();  
    }  
}
```

```
public void myMethod() {  
    System.out.println("Object initialised successfully");  
}  
}
```

# differences b/w classes and objects

	Class	Object
1)	Class is used as a template for declaring and creating the objects.	An object is an instance of a class.
2)	When a class is created, no memory is allocated.	Objects are allocated memory space whenever they are created.  <i>(new)</i>
3)	The class has to be declared first and only once.	An object is created many times as per requirement.
4)	A class can not be manipulated as they are not available in the memory.	Objects can be manipulated.
5)	A class is a logical entity.	An object is a physical entity. ✓
6)	It is declared with the class keyword	It is created with a class name in C++ and with the new keywords in Java.
7)	Class does not contain any values which can be associated with the field.	Each object has its own values, which are associated with it.
8)	A class is used to bind data as well as methods together as a single unit.	Objects are like a variable of the class.

encapsulation

## $\Rightarrow$ Constructors

- In java, a constructor is a block of code similar to method.
- It is a method with no return type
- Imp → It is always called when an object is created using new keyword
- It has same name as of class.

### Syntax

```
public class MyClass {  
    public MyClass() {  
        // initialise an object.  
    }  
}
```

## ⇒ Imp points

- Constructor have same name as class.
- Constructor do not return any value
- Constructor are called only once when an object is created which is not true methods.
- a constructor cannot be abstract, static, final & synchronized.

## → Types of Constructors

- Default Constructor :- which does not accept any parameter
- Parameterized Constructor :- which accepts parameter.

Code

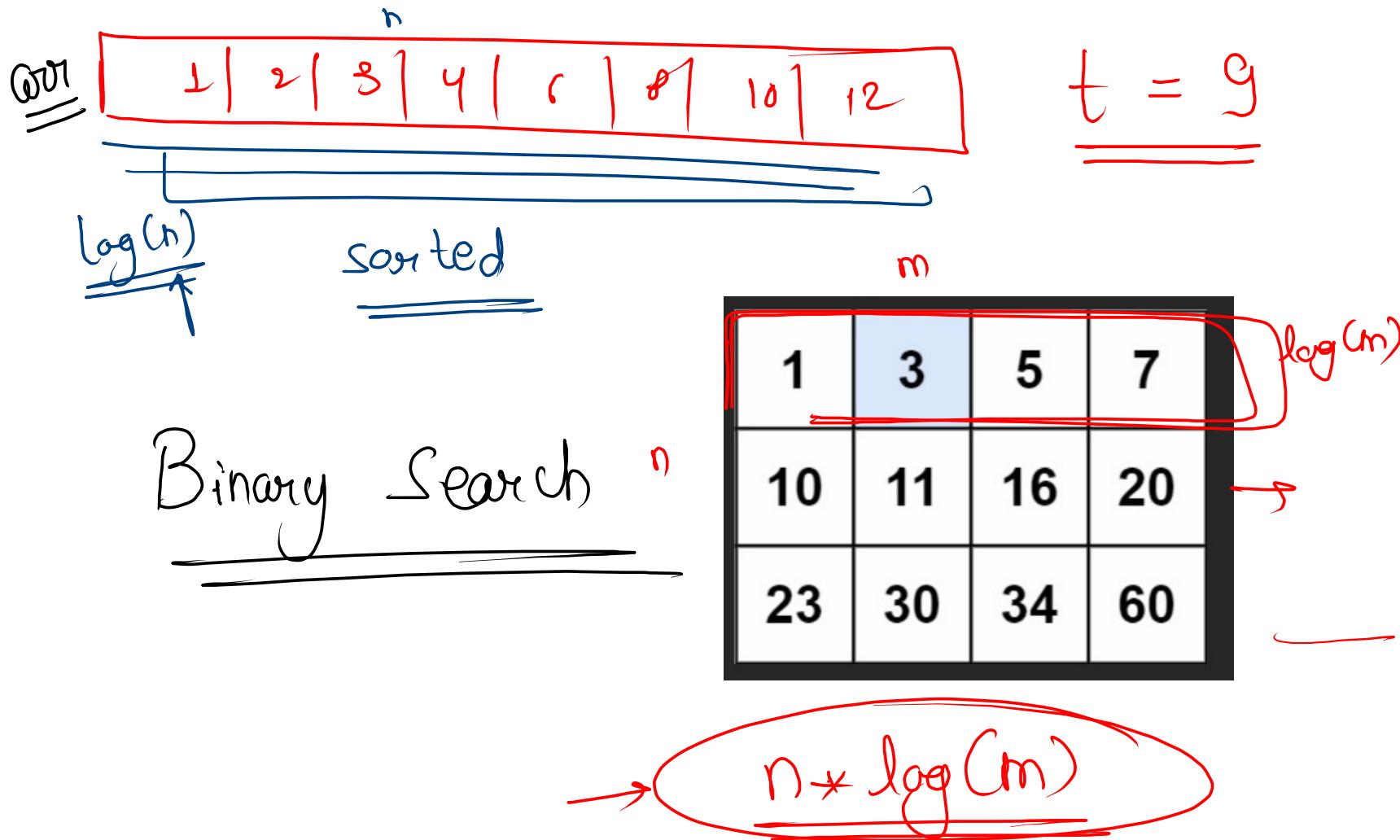
```
public class MyClass {  
    // Fields  
    private int a;  
    private int b;  
    private int age; // not initialised  
    private String name;  
  
    // Default Constructor  
    public MyClass() {  
    }  
  
    // Parametrized Constructor  
    public MyClass(int a, int b) {  
        this.a = a;  
        this.b = b;  
    }  
  
    public MyClass(int b, String name) {  
        this.b = b;  
        this.name = name;  
    }  
  
    public MyClass(int a, int b, int age, String name) {  
        this.a = a;  
        this.b = b;  
        this.age = age;  
        this.name = name;  
    }  
  
    // main method  
    public static void main(String[] args) {  
        MyClass obj = new MyClass(5, 6, 7, "abcd");  
    }  
}
```

constructor  
overloading

## ⇒ Constructor Overload

↳ when we have multiple constructor with same name but different parameters.

## 74. Search a 2D Matrix



To convert 1d to 2d

$$i = \text{idx}/q ;$$

$$j = \text{idx \% q} ;$$

tar =

$$\text{mid} = 5$$

$$si = 0, ei = 11$$

$$\text{mid} = 5$$

curr

0	1	3	5	7
4	10	11	16	20
8	23	30	34	60

ei

curr [i] [j]

curr [mid / m] [mid % m]

```
class Solution {
    public boolean searchMatrix(int[][] arr, int target) {
        int n = arr.length;
        int m = arr[0].length;
        int si = 0;
        int ei = m * n - 1;
        while ( si != ei ) {
            int mid = (si + ei) / 2;

            int curr = arr[mid / m][mid % m];
            if ( curr == target ) {
                return true;
            } else if ( curr < target ) {
                si = mid + 1;
            } else if ( curr > target ) {
                ei = mid - 1;
            }
        }
        return false;
    }
}
```

T.C  
log(m \* n)

