# Introduction to DSA

- Intro. to DSA
- Bits, Bytes, Encodings
- Memory Management in Java

---

## Bits, Bytes, Encoding

Most basic unit of information in a computer (digital device) : a bit

Built by tiny pieces of hardware : "transistors".

IC = integrated circuit

| | State |
|---|---|
| Current flowing through transistor | 1 |
| No Current | 0 |

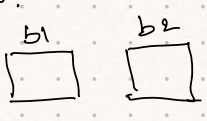

1 bit can represent two states: 0, 1

light on ... 1

light off ... 0

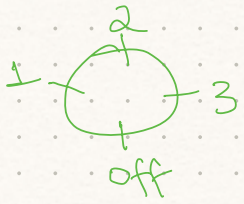
We can use 1 bit to represent the state of a light bulb.

e.g. Two bits:




| b1 | b2 |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

4 possible value (states)




Fan regulator

Fan State

| | | |
|---|---|---|
| Off | 0 | 0 |
| Speed 1 | 0 | 1 |
| Speed 2 | 1 | 0 |
| Speed 3 | 1 | 1 |

Three bits

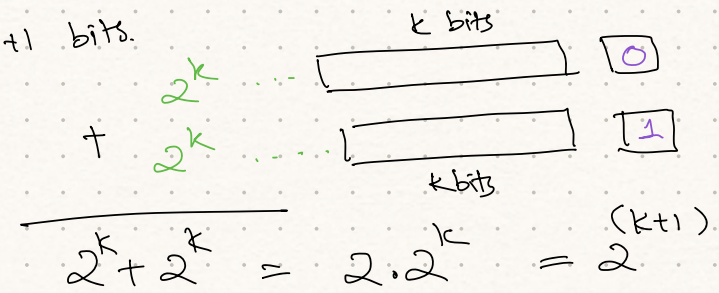
| b1 | b2 | b3 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

8 possible values (states)

Pattern

| # bits | # values (states) that can be represented |
|---|---|
| 1 | $2 = 2^1$ |
| 2 | $4 = 2^2$ |
| 3 | $8 = 2^3$ |
| 4 | $16 = 2^4$ |
| ⋮ | ⋮ |
| k bits | .. ? $2^k$ values. |

k+1 bits.

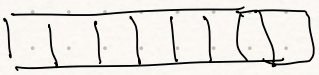



$2^k + 2^k = 2 \cdot 2^k = 2^{(k+1)}$

## Bytes

1 dozen = 12

1 Byte = 8 bits



1 KiloByte (KB) = 1024 Bytes

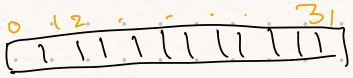
1 MegaByte (MB) = 1024 KB

1 GigaByte (GB) = 1024 MB

1 Tera Byte (TB) = 1024 GB

⋮ ⋮

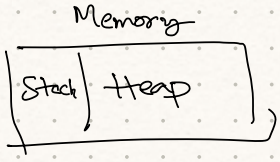
## Primitive Type




| | #bytes | # bits | #values |
|---|---|---|---|
| int | 4 | 32 | $2^{32}$ (diff integers) |
| long | 8 | 64 | $2^{64}$ |
| byte | 1 | 8 | $2^8 = 256$ |
| boolean | — | 1 | $2^1 = 2$ values |

# Memory Management in Java

Two places where Java stores data in a program



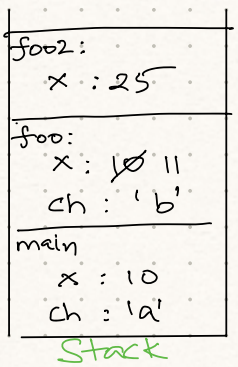

- **Stack** (storing primitive values)
- **Heap** (Store objects)

Every function has its own seperate stack of values.

```
main() {
    int x = 10;
    char ch = 'a';
    foo(x)
}
```

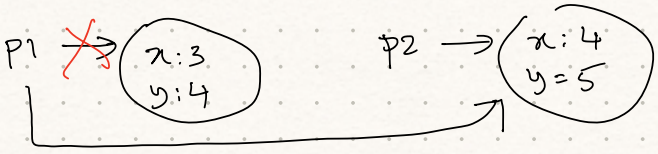
```
foo(int x) {
    x = x+1
    char ch = 'b';
    foo2()
}
```

```
foo2() {
    int x = 25;
}
```

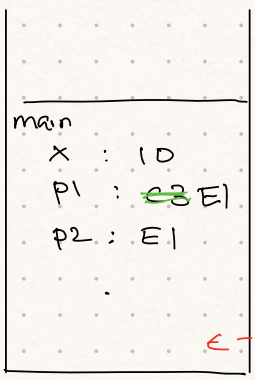


```
main() {
    int x = 10;
    Point p1 = new Point(3,4);
    Point p2 = new Point(4,5);
    p1 = p2; // Assigment of Reference type
}
```

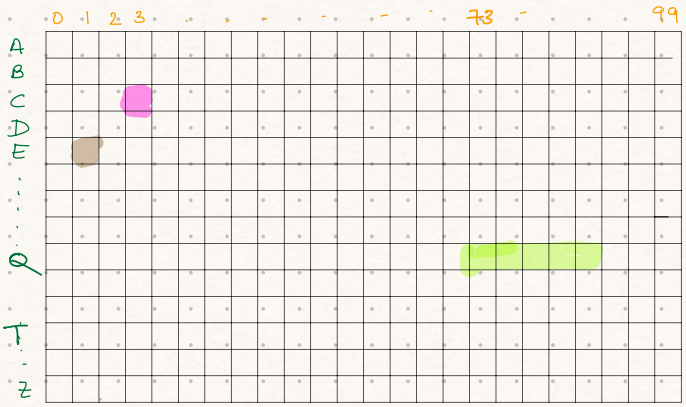
(Both p1 & p2 refer to same object)




"Address" / Location of point object : C3




Stack




Heap

<- - addr (location) of object that p1 refers to.
(reference to the object)