

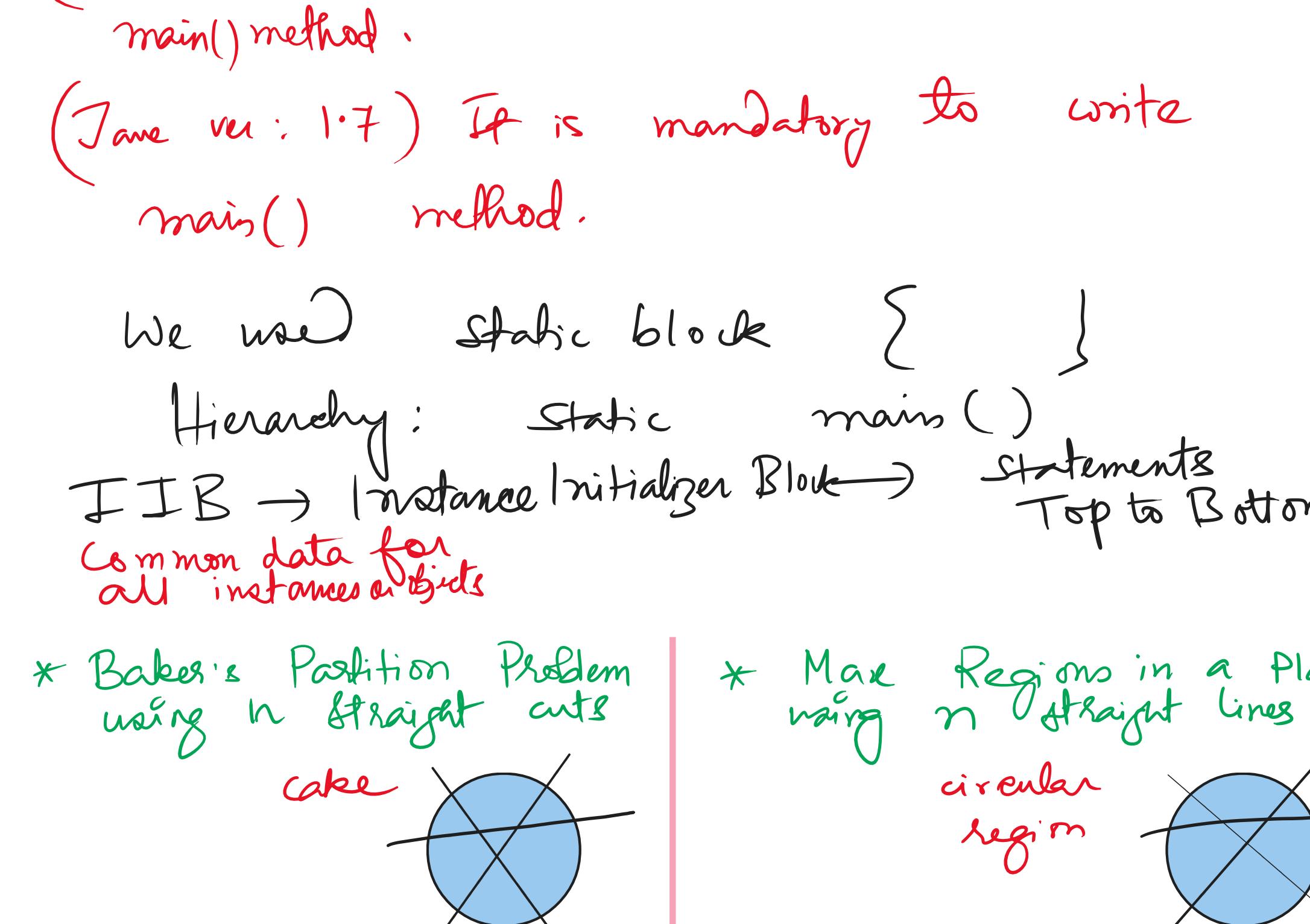
## Concepts that are mandatory for Recruitment / Placement Drives

C	C++	Java	Python
<ul style="list-style-type: none"> <li>* History</li> <li>* Data types</li> <li>* Operators</li> <li>* Conditions</li> <li>* Looping</li> <li>* Functions</li> <li>* Recursion</li> <li>* Pointers</li> <li>* Array Pointers</li> <li>* Enums</li> <li>* Struct &amp; Union</li> <li>* DMA</li> <li>* Files</li> </ul>	<ul style="list-style-type: none"> <li>* History</li> <li>* namespaces</li> <li>* cin // getline</li> <li>* <b>data types: auto</b></li> <li>* STL</li> <li>L Standard Template Library</li> <li>{ stack / queue / map / list / pqr }</li> <li># include &lt;bits&gt; / stdc++.h</li> <li># Virtual functions</li> </ul>	<ul style="list-style-type: none"> <li>* History</li> <li>* WORA <math>\rightarrow</math> C</li> <li>* Structure Anatomy</li> <li>* Wrapper Classes</li> <li>* Boxing / Unboxing int <math>\rightarrow</math> Integer</li> <li>* Files</li> <li>* OOPs</li> <li>*** Exceptions</li> <li><b>Collections Framework Java-util</b></li> </ul>	<ul style="list-style-type: none"> <li>* Membership Operators in &amp; not in</li> <li>* Identity operators is &amp; is not</li> <li>* Walrus Operator :=</li> <li>* Duck Typing</li> <li>* Operator Overloading</li> </ul>

$n_1 = 0$        $1, 2, 3, 4, 1, 2$   
 $n_2 = 0$        $x \oplus A = 1 \wedge 2 \wedge 3 \wedge 4 \wedge 1 \wedge 2 = 7$   
 rightmost set bit =  $\boxed{1}$   
 $\star\star\star [x \text{OR} n_1 - x \text{OR} n_2]$   
 $7 \rightarrow 0111$        $7 \& -7 = 1 \wedge 1 \wedge 3$   
 $1000$        $0111 \& 1001 = \boxed{1}$   
 $+ 10001$        $\boxed{3}$   
 $\cancel{1001} = 0001$   
 $\star\star\star \text{for } (\text{int } i=0, \text{ i} < \text{arr.length}; \text{ i}++) \{ = 0 \}$   
 $\star\star\star \text{if } (\text{arr}[i] \& \text{rsb} = \text{arr}[i])$   
 $\star\star\star \text{else } n_1 = \cancel{\text{arr}[i]} = 1 \wedge 1 \wedge 3$   
 $n_2 = \cancel{\text{arr}[i]} = 4$   
 $2 \wedge 2 \wedge 4$   
 $\cancel{2} \wedge \cancel{4}$   
 $\cancel{2} \wedge \cancel{4}$   
 $\cancel{4}$   
 $\cancel{4}$   
 $\cancel{2} \wedge \cancel{2} \wedge \cancel{4}$   
 $\cancel{2} \wedge \cancel{2} \wedge \cancel{4}$   
 $\cancel{4}$   
 $\cancel{4}$

Static  $\rightarrow$  Stack Memory  
 $\text{int } a = 10;$       value  $\xrightarrow{\text{ptr}} \&$   
 $\text{val}$        $0x1000$   
 $\text{f1.c}$        $\text{f2.c}$   
 $\text{Rn8 folder}$   
 $\boxed{\text{static } x}$   
 $\boxed{\text{extern } y}$

**Storage Classes:**  $\rightarrow$  C & C++



We use static block { }

Hierarchy: static mains()

IIB  $\rightarrow$  Instance Initializer Block  $\rightarrow$  statements Top to Bottom

Common data for all instances

\* Baker's Partition Problem using straight cuts



No of Cuts	Max Pieces
0	$S(n)+1$
1	$\frac{n \times (n+1)}{2} + 1$
2	4
3	7
4	11
5	16

\* Max Regions in a Plane using n straight lines



No of st lines	Max Regions
0	1
1	2
2	4
3	7
4	11
5	16

WORA  $\rightarrow$  Write Once Run Anywhere

Platform Independent  
Portable  
Any O/S

Sanity testing

Why?

(Compiled & Interpreted)

JDK Path  $\rightarrow$  env vars

JRE  $\Rightarrow$  O/S jdk

javac file.java

Byte code ojava.class

OOPS (T-F) B1

Exceptions (S-M) B2

C++ Exceptions (T-W-T-F) B1 XXXX

Files & File Handling (Collections + DSA)