Ex

```
main(){
    for(int i=0; i<a; i++){
        for(int i=0; i<b; i++)
            printf("%d");
    }
    for(int i=0; i<c; i++)
        printf("%d");
}
```

operations:- $(a*b)+c$

$T.C = O((a*b)+c)$

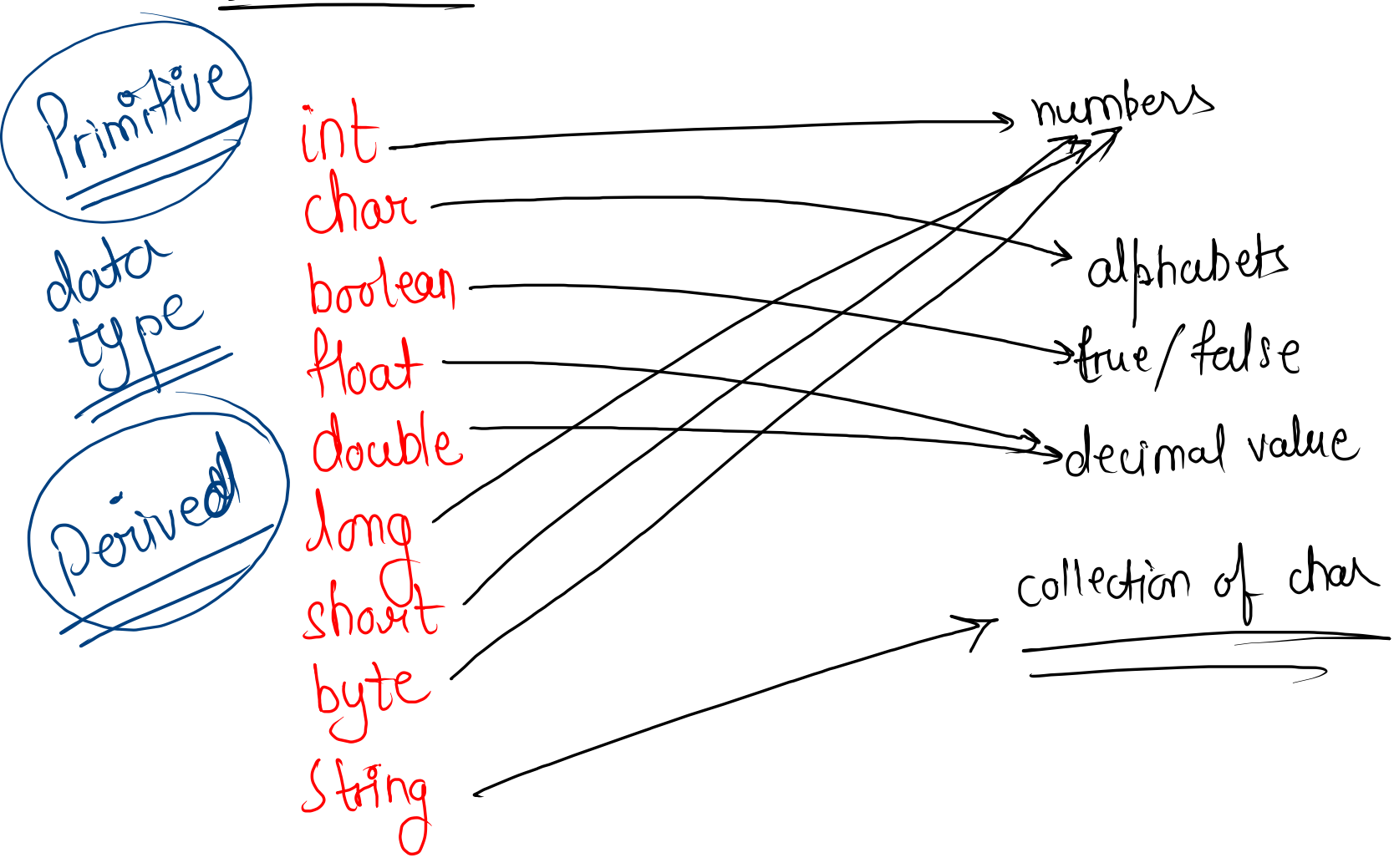
Revision

- Programming Language & Operators
- Variables
- Condition and logical operators
- ★ if else condition (ladder)
- nested if else
- switch case
- ★★ Characters and Strings (typecasting & asciivalue)
- ★ for loops
- while and do while loop
- ★ Patterns (Template)
- Functions (return statement)
- ★ Digit Traversal & Number theory (/10, %10)
- ★★★ Arrays (Printing, updating, searching, upgradation)
- ★ Brute force approach (Permutation & Combination)
- ★ Time Complexity & space complexity

⇒ Operators (special symbols used to perform operations)

- airthematic :- + , − , * , / , %
- logical operation :- && , || , !
- Relational operation :- > , < , >= , <= , == , !=
- assignment operation :- = (right to left)
- Unary operator :- a++ , a-- , ++a , --a
- Ternary operator :- (relation) ? true : false ;

⇒ variables


Primitive
data type
Derived
int
char
boolean
float
double
long
short
byte
String
numbers
alphabets
true / false
decimal value
collection of char

⇒ Conditions

1)
```
if ( condition ) {
    // statement
}
```

---

2)
```
if ( cond1 ) {
} else {
}
```

Notes :-

1) always a single statement can be executed

2) always check from top to bottom

3) if cond" is mandatory & all others are optional

4) we can check 1 or more condition in one statement

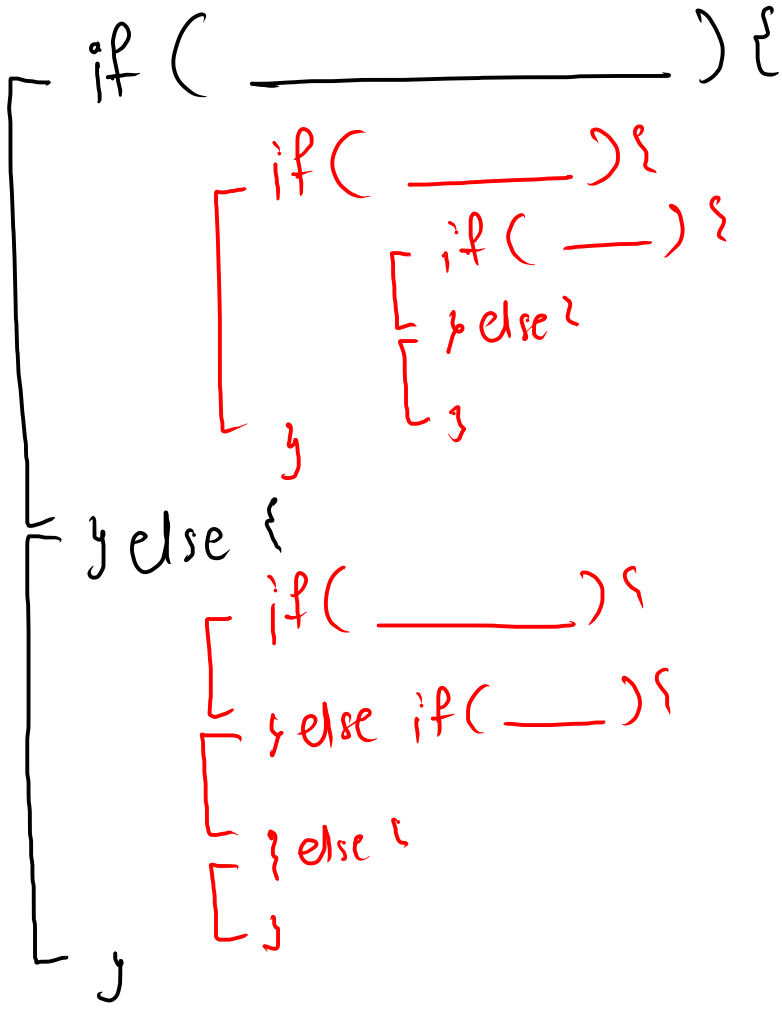
ladders

```
if ( cond1 && cond2 ) {
        ____ s1
} else if ( cond 3 ) {
        ____ s2
} else if ( cond 4 ) {

}
.
.
.
else {
        statement
}
```

if else ladder

nested if else



Nested

syntax switch statement

```
switch ( condition ) {
    Case val1 :
        //statement 1
        break;
    Case val2 :
        //statement 2
        break;
    Case val3 :
        //st 3
        break;
    default :
        //st 4
        break;
}
```

never compulasary

⇒ How to manupulate no. of decimal digits

```java
public class Main {
    public static void main(String[] args) {
        int a = 15;
        int b = 5;
        double ans = a / b;
        System.out.println( String.format("%.5f", ans) );
    }
}
```

⇒ Characters & String

String str = "Geekster classes";

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

inbuilt functions

- str.length() ⟶ 16
- str.charAt(index) ⟶ 'e' (index = 9)
- str.toUpperCase() ⟶ "GEEKSTER CLASSES"
- str.toLowerCase() ⟶ "geekster classes"
- concatenation :- str + str

str1 = "abc"

str2 = "efg"

str1 + str2 = "abcefg"

str2 + str1 = "efgabc"

str1 + 25 = "abc25" → charAt(4) ⇒ '5'

for loops

```
for ( initialisation ; conditions ; upgradation) {
        // statement
}
```

initialisation :- from where to start

condition :- when to stop

upgradation :- how to move

while loop

```
initialisation
while ( conditions ){
        // statement

        upgradation
}
```

Note :- both can be used to solve any question

Imp

ascii values

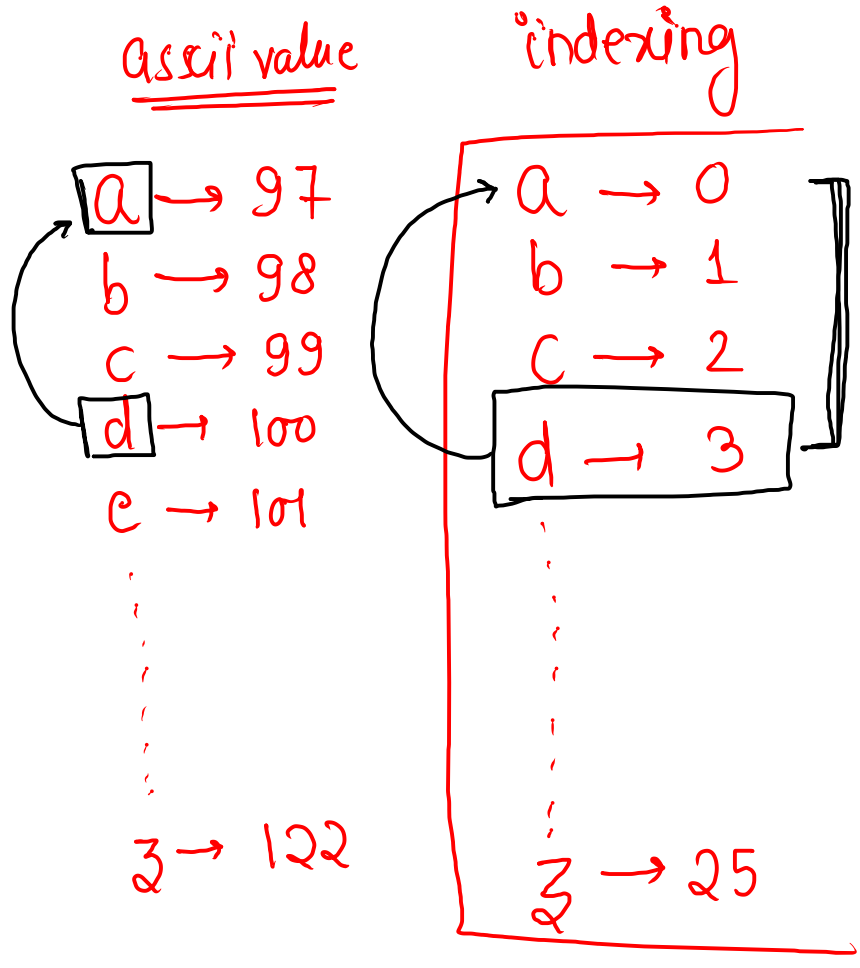
char ch = 'd'

int index = ch - 'a'

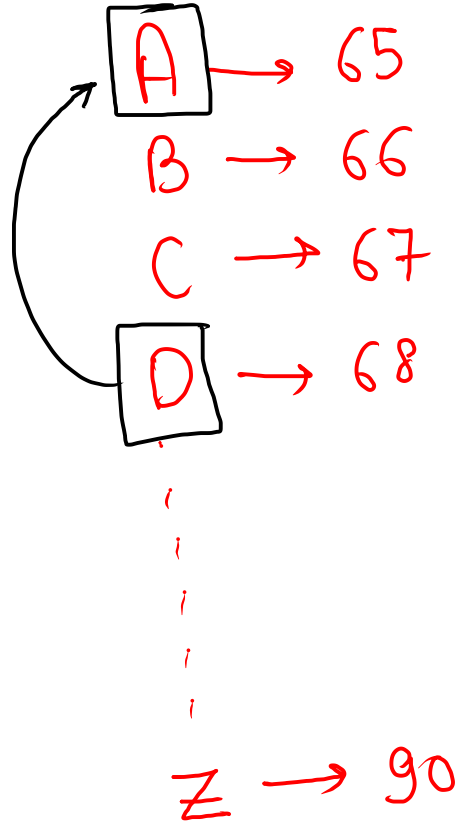
= 'd' - 'a'

= 100 - 97

= 3



```
char ch = 'D'
int idx = ch - 'A'
        = 'D' - 'A'
        = 68 - 65
        = 3
```



char ch = '4'

int idx = ch - '0'

= '4' - '0'

= 15 - 11

int idx = 4