

(Learn Pointers Topic)

- O.S
- DBMS
- Networking
- System Design

Set Vision And Goal

(Self belief)

You are On Right Path

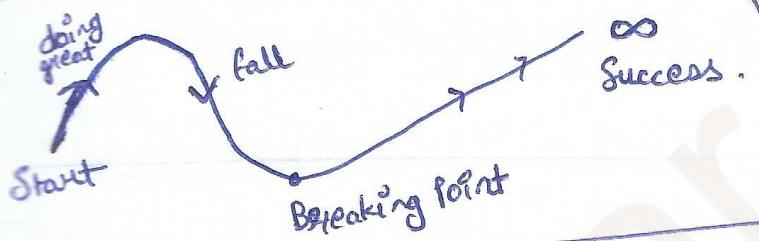
- ↳ Coding will take time
 - ↳ Long term plans
- Good results take time

Solve your own problem.

Too much theory Less practice

Avoid this

Self Practice



Path

1 Sem

Sem

- Language ①
- DSA ②
- Practice (300-400) Qs ③
- Competitive Programming ④
- Development
- ↳ Build Project (real life solving)
- ↳ Build Resume
- ↳ 2nd yr Internship
- Open Source
- ↳ Find Your Niche

• Tips -

Day 1 → Learn (Afternoon)

Day 2 → Practice (Afternoon)



Once Start

No stop



• Detailed notes from Averages

• Practice each question.

• College subject make perfect only till you get good marks in exam.

↳ This course has everything that a tier 1 college has.

↳

* Some important functions

① swap(nums, i, j); for (int i = 0; i < n; i++) {
 for (int j = i + 1; j < n; j++) {
 if (nums[i] > nums[j]) {
 swap(nums, i, j);
 }
 }
 }

② int partition(int arr[], int low, int high) {

 int pivot = arr[low];

 int i = low + 1;

 for (int j = i; j < high; j++) {

 if (arr[j] < pivot) {

 swap(arr, i, j);
 i++;

 }
 }

 return i - 1;

}

 }

 }

Flowcharts → Diagram to represent solution of problem.

- ↳ ① Divide the work in small parts.
- ② Logically solve the small parts.

Ex- Making of tea → Start.

bowl the water



gas stove on



Sugar, tea, leaves, milk



Boil



Exit

• Components

1) Start / Exit -

Shapes

Start

Read N

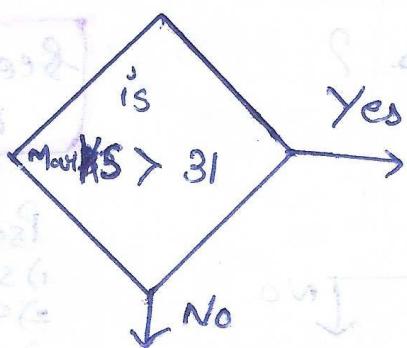
Print Hi

name = Value

variable

Assignment operator

4) Decision -



And these all shapes are connected with

(various) →, ↓

Example of Problem

Sum of two Numbers

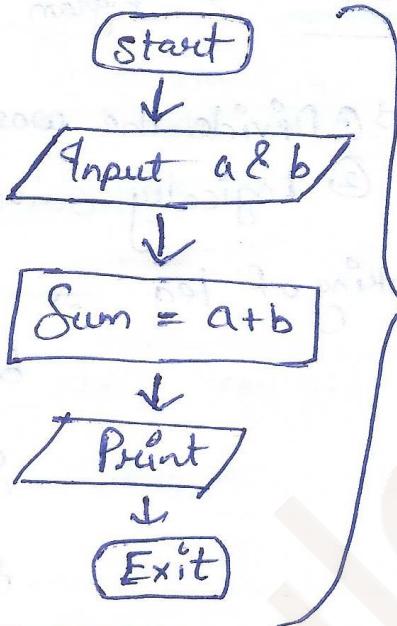
Input

first no., a

Second no., b

Output

Sum of a & b



Calculate Simple Interest

Input

- Principal, P
- Rate, R
- Time, T

Multiplication sign in Java $\rightarrow *$

Output

$$SI = \frac{PRT}{100}$$

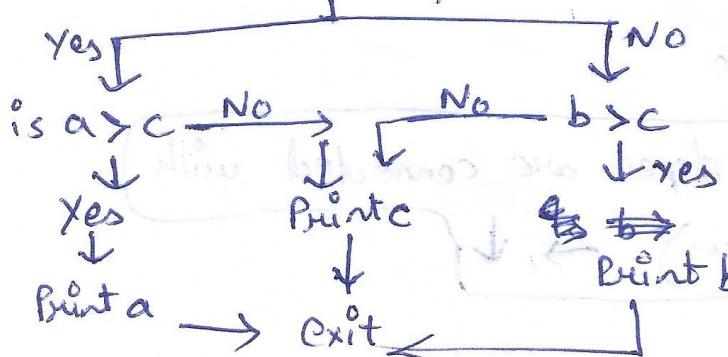
Formulas may not be reminding that's not a problem.

Pseudo Code

- 1) Start
- 2) Input principals (P)
Rate (R)
Time (T)
- 3) Calculate
 $SI = (P * R * T) / 100$
- 4) Print SI (Simple Interest)
- 5) Exit

Find max of 3 numbers ?

- 1) Input a, b, c
- 2) Is $a > b$?



See what your brain think

Pseudo code

- 1) Start
- 2) Input a, b, c
- 3) If $a > b$ do
 - if $a > c$ do
 - Print a
 - else Print c
- else if $b > c$ do
 - Print b
- else Print c

Q Find the number is prime or not?

⇒ Pseudo code

- 1) Start
- 2) Input no. n
- 3) While $\text{div} < n$ do
 - 4) If $n \% \text{div} == 0$ do
 - Print "Not prime"
 - Exit
 - Else $\text{div} = \text{div} + 1$
- 5) Print Prime
- 6) Exit

Non prime Numbers

↳ Can be divide by
the no. bet^n

$$2 \longleftrightarrow (n-1)$$

↳ ex - 6

$$2 \longleftrightarrow (6-1)$$

→ Six can be divided
by 2,3

∴ It is not a prime

Q sum of first n natural Numbers?

⇒ Pseudo code

$$\boxed{n * (n+1) / 2}$$

- 1) Start
- 2) Input no. n

$$3) \text{if } n * (n+1) / 2 = a \text{ then}$$

- 4) Print a

- 5) exit

Sum of 25 scores? (Average)

- ⇒
- 1) Sum = 0, count = 0
 - 2) Enter s.
 - 3) Sum = Sum + s
 - 4) Count = Count + 1
 - 5) If count = 25
 - 6) Av sum/25

- 1) Start

- 2) Input n

- 3) Let value = 1, sum = 0

- 4) While value < n do

$$\text{Sum} = \text{Sum} + \text{Value}$$

$$\text{Value} = \text{Value} + 1$$

- 5) Print

- 6) Exit

If in question asked no. bet^n and 100 then
the no. are $\Rightarrow 10, \dots, 98$

Solution can be found by formulae or
human use.

Variables and Data Types

~~class~~

• Name of our file and public class
should always be same.

Ex -

JavaBasics.java

Public class JavaBasics

Compiler finds the main program first

• Basic first line of java code...

```
1 public class (name of program) {  
2     Public static void main {String args []} {  
3         [All other code is written here]  
4     }  
5 }
```

// Boiler plate code

Out put in Java

System.out.print("Hello World")
↓
function

String

;

Terminator

- ! System → S should be capital
- ! OUT → O should be small
- ! print → P should be small

In → next line

or
{ "Hello\n" }

Print Pattern

```
System.out.println(" * * * "); ← output  
System.out.println(" * * "); ← input  
System.out.println(" * "); ← statement terminator.
```

System.out.println(" * * * ");
System.out.println(" * * ");
System.out.println(" * ");

Output in Java

```
System.out.print("Hello World");
```

↓
function

Variables in Java

- Literal → which have fixed values.

ex - 2, 3, Character a, b, c etc.

- Variables → which can change

2 * (a+b)

↓
Literal

↙
variables

↳ It's a memory in which something is stored.

Data types in Java

Java is a typed language

↳ Here we need to specify int, float

Primitive

- byte
- short
- char
- boolean
- int
- long
- float
- double

Non-Primitive

- String
- Array
- Class
- Object
- Interface

1) byte

Size - 1 byte

↳ In this value can be stored

ex - byte b = 8;

↳ print \Rightarrow 8

256 types can be stored (-128 to 127)

2) char

Size - 2 byte

'a' to 'z' & (A-Z) (symbols)

↳ ex -

char ch = a;

↳ single character can be stored

3) boolean

Size \rightarrow 1 byte

(true/false)

↳ Only two values

True / False

ex -

boolean var = false;

4) float

Size - 4 byte

↳ Decimal number can be stored

ex -

float price = 10.5;

5) int

(-2 billion, 2 billion)

Size - 4 bytes

↳ If we want to store a integer

ex -

int number = 25;

6) long \rightarrow A number of integer

8 byte

7) double \rightarrow A long number of ~~decimal~~ a decimal number

size - 8 byte

8) short

Size \rightarrow 2 byte

↳ 1 number can be stored

ex -

short n = 240;

Question - Sum of a + b

\Rightarrow int a = 10;

int b = 5;

int sum = a + b

• Keep the name of variable Logical

Comments in Java

↳ statements that does not execute in java

1) // it will be ignored } single line comment

2) /*

 = }

 */ } Multiline Comment

↳ Compiler ignores.

Input in Java

```
import java.util.*;  
Scanner sc = new Scanner(System.in)
```

String input = sc.next();
 ↑
 variable

• make a class

Scanner

• util is a package

This next captures
till only next 'space'
in input

• nextLine → captures till next line

For taking input of a number?

⇒ int number = sc.nextInt();

System.out.println(number);

• For floating value?

⇒ float price = sc.nextFloat();

System.out.println(price);

• And more methods

→ nextByte

→ For byte type data

→ nextDouble

→ nextBoolean

→ nextShort

→ nextLong

Sum of a and b (By taking input)

Question

Scanner →

```
int a = sc.nextInt();  
int b = sc.nextInt();  
int sum = a+b;  
System.out.println(sum);
```

Program in PC

Question

Multiplication of a and b

```
int a = sc.nextInt();  
int b = sc.nextInt();  
int product = a * b;
```

```
System.out.println(product);
```

Program
in PC

Question

Area of circle

Scanner

```
int r = sc.nextInt();  
int d = sc.nextInt();
```

```
float area = 3.14 * r * r
```

```
System.out.println(area);
```

Program in PC

* If we add a (f) with out decimal value it becomes floating number.

Example → 3.14f

And if not then it will be a double

Type Conversion / Widening / implicit conversion

↳ happen only betⁿ compatible types

`byte -> short -> int -> float -> long -> double`

↳ and the ^{type} should be greater than the type that is converted.

Type Casting / Narrowing conversion / explicit

→ Force full type conversion

`float a = 25.99`

`int b = (int)a;`

`System.out.println(b);`

Output
= 25

Character can also be converted into number

`a = 97`

`b = 98`

`c = 99`

`d = 100`

Type Promotion in expression

↳ Java automatically promotes

`byte, short or char` → `Int`

when evaluating an expression.

and when
`int, float, long, double`
are written together
it get converted
into double

`int, float, long → double`
i.e.

Ex-

`char a = 'a';`

`char b = 'b';`

`System.out.println((int)(a));`

`System.out.println((int)(b));`

`System.out.println((b-a));`

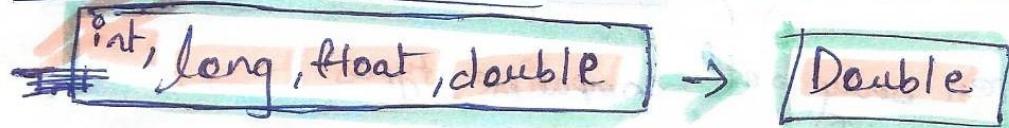
Output

97

98

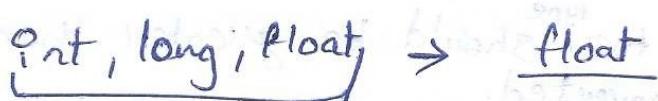
1

- During expression. Automatic conversion by java



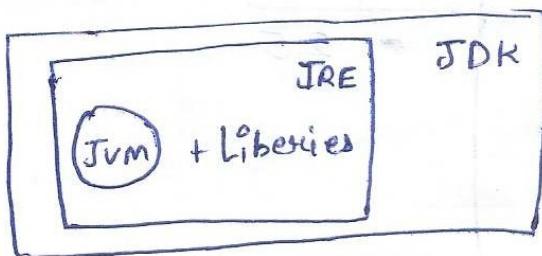
Its get converted into the greatest one

Ex -



How is our code is running?

JRE → Java runtime environment



Source code (.Java)

Compiler

Byte Code
.class

Java Virtual Machine
(JVM)

Native Code

Java is a portable language

C++ is not

* Identifiers can start with any variable

including (-) and ~~(\$)~~.

Operators in Java

add → +

Subtract → -

Multiply → *

divide → /

operands
operator
expression

$$\text{Sum} = a+b$$

Types of operators

1) Arithmetic operators

Binary (2 operands)

- + → A+B
- - → A-B
- * → A*B
- / → A/B

(Modulo) % → A%

For remainder

ex - $10 \% 5 = 2$

ex - `int a = 10;`

`int b =`

Unary (Increment)

- $a = a+1 \rightarrow a++$ or $+a$
- $a = a-1 \rightarrow a--$ or $--a$

(Decrement)

++a → Pre Increment

- 1) First value changed
- 2) then it is used

a++ → Post increment

- 1) First it is used
- 2) then value is increased.

2) Relational Operators

- == To check values are equal or not

ex - $a=b \rightarrow \text{False}$

$(5)(10)$

We get result as ~~True~~ True and False

- != not equal to

> Smaller than

< Greater than

\geq Less than equal to

\leq greater than equal to

3) Logical Operators

True $\rightarrow T$

False $\rightarrow F$

i) Logical AND $\Rightarrow \&&$

True $\&&$ True $\rightarrow T$

F $\&&$ F $\rightarrow F$

T $\&&$ F $\rightarrow F$

F $\&&$ T $\rightarrow T$

ii) Logical OR $\Rightarrow \mid\mid$

T $\mid\mid$ F $\rightarrow T$

F $\mid\mid$ T $\rightarrow T$

F $\mid\mid$ F $\rightarrow F$

T $\mid\mid$ T $= T$

iii) Logical NOT $\Rightarrow !$

!T $\rightarrow F$

!F $\rightarrow T$

4) Assignment Operator

* These are faster than normal operators

i) $=$ ex - $A = B$

ii) $+=$ ex - $A = A + 10 \rightarrow A+ = 10$

iii) $-=$ ex - $A = A - 10 \rightarrow A- = 10$

iv) $*=$ ex - $A = A * 2 \rightarrow A* = 2$

v) $/=$ ex - $A = A / 2 \rightarrow A/ = 2$

Operator Precedence → which will execute first.

ex- int $x = 3 * 4 - 1$

→ Here $x = 11$
not 9.

- 1) ~~$\boxed{3} \boxed{*} \boxed{4} \rightarrow \boxed{+} \boxed{4} \rightarrow \boxed{-} \boxed{1}$~~) $\boxed{++} > \boxed{-}$
- 2) ~~$\boxed{3} \rightarrow \boxed{*} \rightarrow \boxed{4} \rightarrow \boxed{+} \rightarrow \boxed{-} \rightarrow \boxed{1}$~~ Left to right
- 3) ~~$\boxed{*} > \boxed{/} > \boxed{\%}$~~

4) $\boxed{+} > \boxed{-}$

5) $\boxed{<<} > \boxed{>>} > \boxed{>>>}$

6) $\boxed{<} > \boxed{<} > \boxed{<=} > \boxed{>} > \boxed{>=}$ instance of

7) $\boxed{==} > \boxed{!=}$

8) $\boxed{>>=} > \boxed{>=} > \boxed{<<=} > \boxed{||=} > \boxed{^=} > \boxed{&=} > \boxed{/=} > \boxed{*=}$

in a line have same precedence

• If in a line have same precedence
the expression is evaluated to according
to its associativity (either left → right or
right → left)

changes in

Notation

Notes: PRACTICE

↳ functionality (position) = priority

Smart quotes ↴ ↴

= global ↴ ↴

= global ↴ ↴

i.e.: ?(?) = repeat n times

? ← fugit

"fugit": "new" \$(o == 10^3) = sqrt point

Conditional Statements

1) if - else Statements : $\Rightarrow \text{if}(\text{condition})\{\dots\}$

i) Program \rightarrow Print largest numbers.

ii) Program \rightarrow Print number is odd or even.

$\}\text{else}\{\dots\}$

2) else if Statements.

$\Rightarrow \text{if}(\text{condition } 1)\{\dots\}$

$\text{else if}(\text{condition } 2)\{\dots\}$

$\text{else}\{\dots\}$

- Parenthesis $\rightarrow ()$
- Curly braces $\rightarrow \{\}$

i) Program \rightarrow Print largest of ~~of~~ three numbers.

Ternary Operator

\hookrightarrow 3 operands.

variable = condition ? Statement 1 : Statement 2;

Nahitoh

true

if this is true
the variable = 1
or variable = 2

Ex -

boolean larger = ($5 > 3$) ? 5 : 3 ;

Output \Rightarrow 5

Ex -

String type = ($5 \% 2 == 0$) ? "even" : "odd";

Program → student will pass or fail?



Switch Statement

```
switch (variable) {  
    case case 1:  
        // code  
    case 2:  
        // code  
    case 3:  
        // code  
        break;  
    default:  
        // code  
}
```

↳ If variable is 1 case one will execute.

↳ If no case is true default execute.

To break the switch.

↳ It can be written after any case to break switch from there.

↳ You can either use int for variable or char or float etc.

#Program o Calculator

Leap year or not?

→ If the year is divisible by -

4 and 100 → Leap year

4 and 400 → Leap year.

108 BY

10801 is

for tipib tool two shot of a
old fi shivib medium

to tel ei rebinishe bren

Loops (Flow control)

1) while Loop :-

```
while (condition){
```

```
//do something  
}
```

Q. Print no. from 1 to 10 ?

Q. Print sum of 1(n) natural no.?

2) for Loop :-

```
for (initialisation; condition; updation){
```

```
//do something  
}
```

ex-

```
for(int i=1; i<=10; i++)
```

{ System.out ("Hello world!");
}

Q. Print square pattern?

Print Reverse of a number

ex n = 10899 → 99801

- To take out last digit of any number divide it by 10 and remainder is last digit

To take out last digit \rightarrow $\text{Num} \% 10$

To ~~remove~~ remove last digit \rightarrow $\text{Num}/10$

Take out the reverse of given number.

$\Rightarrow \text{while}(a > 0) \{$

$$\text{lastdigit} = (a \% 10);$$

$$\text{rev} = (\text{rev} \times 10) + \text{lastdigit}$$

do-while loop

do {

// do something

} while(condition);

Break Statement

break;

\hookrightarrow to get out from a loop.

Q. Break statement when user enters multiple of 10?

{ { i(*) } if you need to keep the loop running
add condition are true. }

Continue Statement

continue;

\hookrightarrow To skip an iteration.

Q. Display all numbers but not multiple of 10.

Q. Check the no. is prime or not?

* If any variable written inside the loop it is only valid inside the loop. Not in the whole program.

Nested Loops

1) Print star pattern ?

x
xx
xxx
xxxx

→
for(int line=1; line<=4; line++){
 for(int star=1; star<=line; star++){
 System.out.print("*");
 }
}

- ① lines(4)
4 outer loop → 4 times
- ② Number of Times
↳ inner loop
If we ith line
then its i times.
- ③ What to print?
→ " * "

2) Inverted star pattern ?

→
xxxx
xxx
xx
x
→ for(int line=1; line<=4; line++)
 { for(int star=1; star=4-line; star++){
 System.out.print("*");
 } }

3) Half pyramid pattern ?

1
1 2
1 2 3
1 2 3 4

Character is initialized using single quotation.

ex- `char = 'A';`

Functions

↳ A block of code
(reserved name of function)

↳ `public static void main(String args [])`

↳ return type → `void`
↳ empty → `String args []`
↳ argument → array

↳ returnType^{name} () {
 // body
 return statement;
}

→ making a new function to print ~~Hello~~ world!

↳ `public static void printHelloWorld () {`

↳ return type → `void`
↳ name → `printHelloWorld ()`

↳ `System.out.println("Hello")`

To call a function.

→ `public static void main (String args []) {`

`printHelloWorld ();`

}

We also need to write return

but If our type is void No need to write return.

Methods → Function that we write inside the class.

o Functions with parameters.

~~return Type name~~

↳

↳ return Type name (type param₁, type param₂);
 //body
 return statement;
}

↳ In parameters we have to show which input we have taken

ex- int a, int b.

4) Two functions can have same name for a variable but they both have different work.

o Functions with parameters.

Parameters.

Ex- public static int sum1 (int num1, int num2) {

 int sum = num1 + num2;
 return sum;

 public static void main (String args[]) {

 Scanner sc = new Scanner (System.in);

 Scanner int a = sc.nextInt();

 - b = -

 int sum = sum1 (a, b);

Arguments

 Sys0 (sum);

Formal parameters
(Parameters)

Actual parameters
(Arguments)

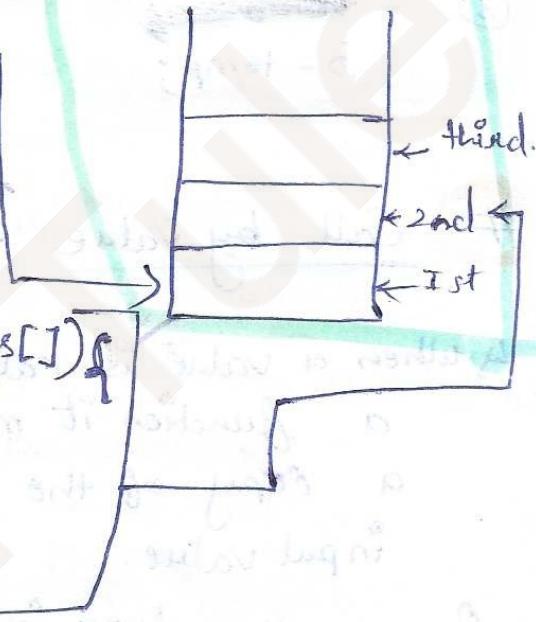
↳ During definition of function

↳ During Calling of function.

= What happens in memory?

→ Public static int sum1(...){
int a=...;
int b=...;
return sum;

Call Stack



② It will occupies the memory

Public static void main(String args[]){
Scanner -
int a -
int b -
sum = sum1(a, b);

③ It will occupies the memory over the first

④ As the program executes in main function memory

↳ Values of int a and b is saved

⑤ In function sum1

↳ sum of a and b is stored

⑥ After return function.

↳ the value is returned to main function

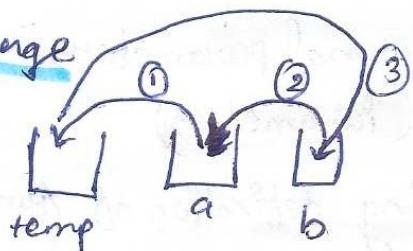
⑦ Now the sum1 function memory will be deleted (empty)

⑧ And after output

↳ main function memory will be deleted.

Swap the values / Exchange

⇒
int a = 10
int b = 5



//swap

- (1) int temp = a;
- (2) a = b;
- (3) ~~b = temp;~~

Used in Java

Call by value → A copy of input value is used in all the functions.

↳ When a value is called in a function it makes a copy of the original input value.

So if return is executed the stack get deleted of that function.

Call by Reference → original value is passed in the function.

↳ Used in C++

Question

① Find product of a & b?

② Find factorial of a number?

③ Binomial coefficient $C_n = \frac{n!}{r!(n-r)!}$

For using square root Use Math Class

⇒ **Math.sqrt()**

- Inbuilt methods vs. User defined methods
- ↳ Already made
 - ex → `sc.nextInt()`
 - ↳ Manually written

Function Overloading

- ↳ We can have multiple ~~function~~ functions with same name and with different parameters
- ↳ Difference in that functions
 - type of parameters.
 - ~~No. of parameters.~~

- ↳ It only works when only parameters are different

Function overloading using parameters

- Q → Function overloading using parameters?
- Q → Function overloading using Data types?
- Q → Check a number is prime or not?

↳ Optimised method.

```
public static boolean prime(int n) {
    for (i=2; i<sqrt(n); i++) {
        if (n%i==0) {
            return false;
        }
    }
    return true;
}
```

Add 2 as an exception.

- Q → Check prime or not - optimised

Q. Print all the numbers that are prime in the range?

Convert from Binary to Decimal

• Decimal number system $\rightarrow 1, 2, 3, 4, 5, 6, 7, 8, 9$

• Binary number system $\rightarrow [1, 2]$

| <u>Decimal</u> | <u>Binary</u> |
|----------------|---------------|
| $(0)_{10}$ | $(0)_2$ |
| $(1)_{10}$ | $(01)_2$ |
| $(2)_{10}$ | $(10)_2$ |
| $(3)_{10}$ | $(11)_2$ |
| $(4)_{10}$ | $(100)_2$ |
| $(5)_{10}$ | $(101)_2$ |
| $(6)_{10}$ | $(110)_2$ |
| $(7)_{10}$ | $(111)_2$ |
| $(8)_{10}$ | $(1000)_2$ |

$B \rightarrow D$

ex - 101

↓ 1 0 1
 $\uparrow \times \uparrow \times \uparrow \times$
 $2^2 + 2^1 + 2^0$
 $\Rightarrow 4 + 0 + 1 = 5$

Code. \rightarrow ex - 1010

~~int pow = 0, pow++~~

~~int lastdigit = n % 10~~

~~Lastdigit = 1~~

~~dec = 0~~

$dec = dec + [lastdigit * 2^{pow}]$

* $\text{Math.pow}(a, b) \Rightarrow a^b$

\Rightarrow PseudoCode

```
int pow = 0
int deciNum = 0
(Num1){
```

while (Num1 > 0) {

int lastdigit = Num1 % 10;

deciNum = deciNum + (lastdigit * (int) Math.Pow(2, Pow));

Num1 = Num1 / 10; }

Pow++;

Q. Convert binary \rightarrow Decimal

Convert Decimal to Binary

Ex - 7

$$\begin{array}{r} 2 \\ \hline 7 \\ 2 \\ \hline 3 \\ 2 \\ \hline 1 \\ 2 \\ \hline 0 \end{array}$$

Quotient

Remainder $\rightarrow 111$

$$B \quad (7)_0 \rightarrow (111)_2$$

Ex - 8

$$\begin{array}{r} 2 \\ \hline 8 \\ 2 \\ \hline 4 \\ 2 \\ \hline 2 \\ 2 \\ \hline 1 \\ 2 \\ \hline \end{array}$$

Remainder
1000

$$8 \rightarrow 1000$$

int pow = 0;
int binNum = 0;
while (num > 0) {

binNum = binNum

binNum = binNum + (binNum % 2) * (int) Math.pow(10, pow);

pow++;

num = num / 2; }

System.out.println(binNum);

Q. Convert Decimal to binary?

while (n > 0) {

① divide by 2

↳ rem

② bin = bin + rem * 10

Scope → If we made a variable then

class scope { } where we can use that variable.

Method Scope → In side a function,

for (i=1) { i only works here }

→ A line where var made

After that everywhere in that function.

Block Scope

{ it will work only here }

If var made here.

Math Class

Ex - $x = 2$
 $y = 4$

- 1) $\text{Math.min}(x, y)$ \rightarrow It will print the minimum number
 $\Rightarrow 2 \quad \Rightarrow 2$
- 2) $\text{Math.max}(x, y)$
 $\Rightarrow 4$
- 3) $\text{Math.sqrt}(y)$
 $\Rightarrow \sqrt{4} \Rightarrow 2$
- 4) $\text{Math.pow}(x, y)$
 $\Rightarrow 2^4$
- 5) $\text{Math.avg}(x, y)$
 $\therefore \frac{2+4}{2} = 3$
- 6) $\text{Math.abs}(a)$ \rightarrow
 - ↳ If a is non negative a is returned;
 - ↳ If a is negative the negation of the argument returned.
 - ↳ If $+0, -0 \rightarrow +0$ returned.
 - ↳ If $\infty \rightarrow +\infty$ returned.

Advanced Patterns

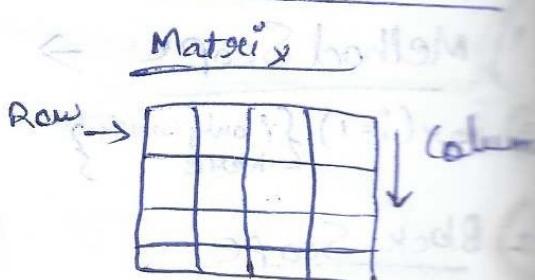
1)

| | | | | |
|---|---|---|---|---|
| * | 2 | 3 | 4 | 5 |
| * | * | * | * | * |
| * | | * | 2 | |
| * | | * | 3 | |
| * | * | * | * | 4 |

boundary

new $\rightarrow 1, 4$

Column $\rightarrow 1, 5$



1) total no. of lines = Rows

Loop outer (1 to 4)

4x4

Q. Print ("*") if row = 1 or 4,

rectangle

Print ("**") if column = 1 or 5,

Print (" ") if not any condition.

| | | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1 | | X | X | X | X |
| 2 | | X | X | X | X |
| 3 | | X | X | X | X |
| 4 | X | X | X | X | X |
| | X | X | X | X | X |

| | | | | | |
|--|--|---|---|---|--|
| | | | | X | |
| | | | X | X | |
| | | X | X | X | |
| | | | X | X | |

// lines

```
for (i=1; i<=n; i++) {
```

n spaces

```
for (j=1; j<=(n-i); j++) {
```

```
    sys0 (" "); } }
```

// star

```
for (j=1; j<=n; j++) {
```

```
    sys0 ("*"); }
```

```
sys0 ("println");
```

1) First
break down
the problem

2) Make a logic

3) Pseudocode

4) Program

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | |
| 1 | 2 | 3 | | |
| 1 | 2 | | | |
| 1 | | | | |

lines = n = 5

to print i=1, i to 5

i=2 1 to 4

1 - 3

1 - 2

```
for (i=1; i<=n; i++) {
```

```
    for (j=1; j<=(n-i)+1; j++) {
```

```
        sys0 (j); }
```

```
} sys0 ("println");
```

(i-a)
(i-a)
(i-a)

| | | | | | | |
|---|---|---|---|---|---|---|
| X | O | O | O | O | O | X |
| X | X | O | O | O | X | X |
| X | X | X | O | O | X | X |
| X | X | X | X | O | X | X |
| X | X | X | X | X | X | X |

Q. FLOYD's Triangle

| | | | |
|---|---|---|----|
| 1 | | | |
| 2 | 3 | | |
| 4 | 5 | 6 | |
| 7 | 8 | 9 | 10 |

int a = 1;

for { $i=1; i \leq n; i++$ } {

for { $j=1; j \leq i; j++$ } {

sys0(a);

a++; }

Q

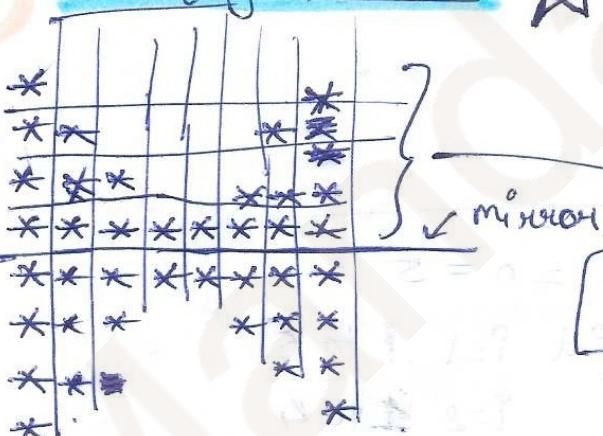
| | | | | | |
|---------|---|---|---|---|------------------------------|
| 1 | 1 | | | | $i=1 \rightarrow 1$ |
| (2,1) 0 | 1 | 1 | | | $i=2 \rightarrow 0, 1$ |
| (3,1) 1 | 0 | 1 | 1 | | $i=3 \rightarrow 0, 1, 0, 1$ |
| (4,1) 0 | 1 | 0 | 1 | 1 | |

for ($i=1; i \leq n;$

if ($i+j$) = even \rightarrow sys0(1)

if ($i+j$) = odd \rightarrow sys0(0)

Q. Butterfly Pattern



for ($i=1; i \leq n; i++$) {

for ($j=1; j \leq i; j++$) {

for

for

I)

$n=4$

for ($i=1; i \leq n; i++$)

Start + space + star

~~for (j=1; j <= 2*(n-i); j++) {~~

① $i=1=4$

~~sys0(" * ")~~

for ($j=1; j \leq 2*(n-i); j++$)

~~else { sys0(" * "); }~~

x 0 0 0 0 0 0 x
x x 0 0 0 0 x x
x x x 0 0 x x x
x x x x x x x x

start

$i=1$

Space

$6 = 2 \times (n-i)$

$4 = 2 \times (n-i)$

$2 = 2 \times (n-i)$

$0 =$

for first half ✓

Outer loop

Second half \rightarrow ~~for ($i=n$; $i>=1$; $i--$) {~~
 ~~sys0 ("*")~~

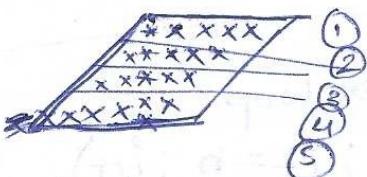
Inner loop same

logic

~~for ($i=n$; $i>=1$; $i--$) {~~

For a mirror image just invert the outer loop

Solid Rhombus Pattern



Outer loop

for ($i=1$; n times)

→ some space \rightarrow star

| | | |
|---------|---|-----------------|
| $i = 1$ | 4 | 5 |
| $i = 2$ | 3 | \rightarrow 5 |
| $i = 3$ | 2 | \rightarrow 5 |
| $i = 4$ | 1 | \rightarrow 5 |
| $i = 5$ | 0 | \rightarrow 5 |

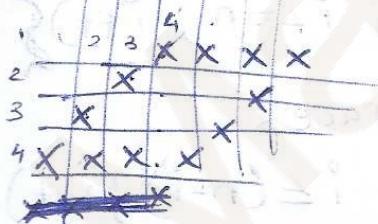
//5 spaces

for ($j=1$; $j \leq (n-i)$; $j++$)

 sys0 (" ")

 nextline;

Hollow Rhombus Pattern



| | space | star |
|---------|---------|-------------------------------|
| $i = 1$ | $(n-i)$ | ($i=0$) |
| $i = 2$ | $(n-i)$ | ($i=0$) |
| $i = 3$ | $(n-i)$ | ($i=0$) |
| $i = 4$ | $(n-i)$ | ($i=0$) |

Hollow rectangle



when $i=1$ || $i=4$

for ($i=1$; $i \leq n$; $i++$) {

 and $j=1$ || $j=4$

 space
 for ($j=1$; $j \leq (n-i)$; $j++$) { sys0 (" "); }

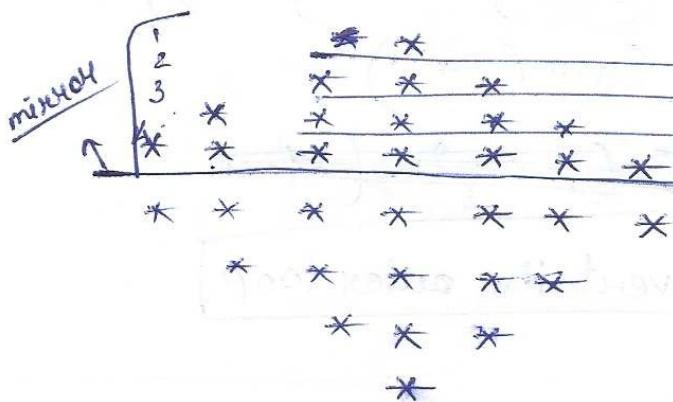
Hollowrectangle

 for ($j=1$; $j \leq n$; $j++$) { if ($i=1$ || $i=n$ || $j=1$ || $j=n$) {

 else (sys0 (" "));

 j = n } { sys0 (*) } }

Diamond Pattern



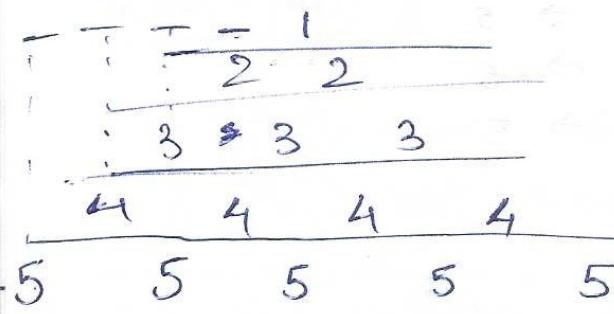
let $n=4$

~~Space + star~~

| | | | |
|-------|-----------------|-------|--------------|
| $i=1$ | $\boxed{[n-4]}$ | $[i]$ | $(2(i-1)+1)$ |
| $i=2$ | $\boxed{[n-5]}$ | 3 | $2(i-1)+1$ |
| $i=3$ | $\boxed{[n-3]}$ | 5 | $2(i-1)+1$ |
| $i=4$ | $\boxed{[n-9]}$ | 7 | $2(i-1)+1$ |

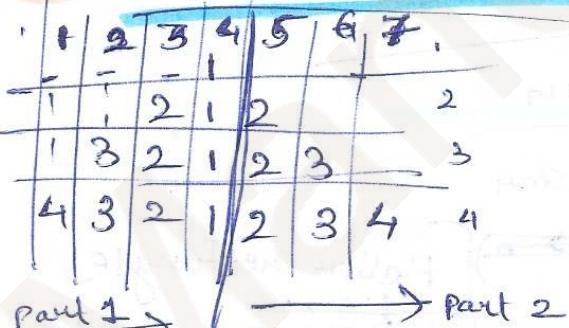
odd number

Number Pyramid pattern



outer loop
 $\rightarrow (i=1; i \leq n; i++)$
~~# spaces~~
 $\rightarrow (\underbrace{\quad}_{j=1} \quad \quad \quad =$
 $(j=1; j \leq (n-i); j++)$
 $+ \# number$
 $\text{for}(j=1; j \leq i; j++)$
 $\text{sys0}(^{\circ}i);$

* * Palindromic Pattern



Part 1 → → Part 2

Part 1 till here
~~# num~~

$\boxed{[(j=1; j \geq 1; j--)]} \quad \boxed{\text{sys0}(^{\circ}j)}$

③ // Part 2

$\boxed{[(j=2; j \leq i; j++)]}$

$\text{sys0}(j)$

outer loop
 $(i=1; i \leq n; i++)$
① ~~# // Spaces~~
 $(j=1; i \leq (n-i); j++)$
~~# // Num~~
~~for(j=1; j < (n-i); j++)~~
~~sys0(^{\circ}j);~~
~~j++~~