(Learn Pointers topic) • O.S
• DBMS
• Networking
• System Design

Set vision And Croal

(Self belife)

You are On Right Path ----

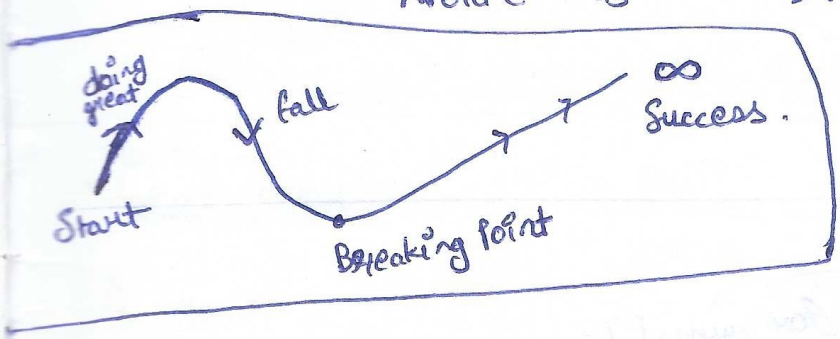
↳ Coding will take time
↳ Long term plans
} Good reasults take time

Solve your own problem.

Two much theory Less pratice

Avoide this

• Self Pratice



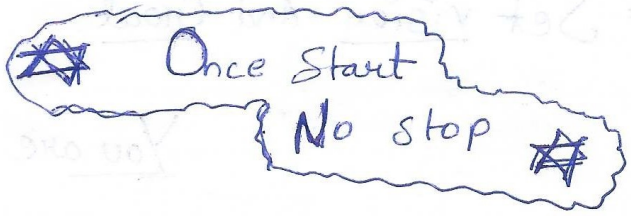
Path 1 Sem

→ Language (1)
→ DSA (2)
→ Pratice (300-400) Qs (3)
→ Competetive Programing
→ Development
} (4)

Sem
↳ Build Project (real life solving)
↳ Build Resume
↳ 2nd yr Internship
Open Source
} or
↳ Find Your Niche

- Tips -

Day 1 → Learn
Day 2 → Pratice



- Detailed notes from Arrays
- Pratice each question.
- College subject make perfect only till you get good marks in exam.

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

↳

---

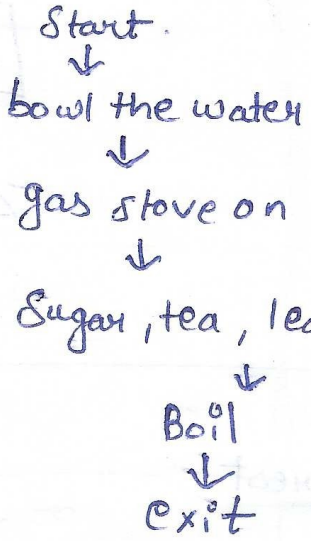
* Some important functions

(1) swap(nums, i, j); for nums[];

Flowcharts → Digram to represent solution of problem.

↳ ① Devide the work in small parts.
② Logically solve the small parts.

ex- Making of tea →

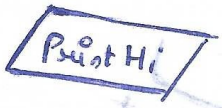


- Components

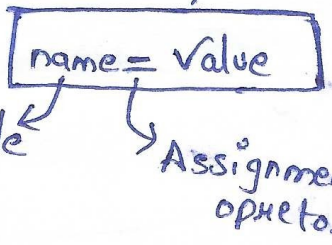
Shapes

1) Start/Exit -

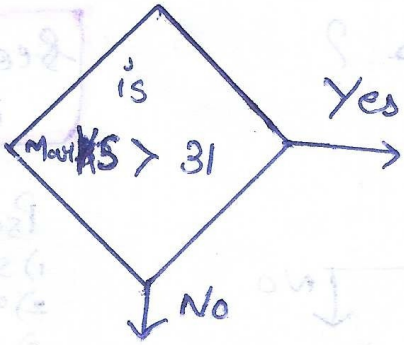
2) Input/Output -



3) Process -



4) Decision -



And these all shapes are connected with (arrows) →, ↓

Example of problem

- Sum of two Numbers

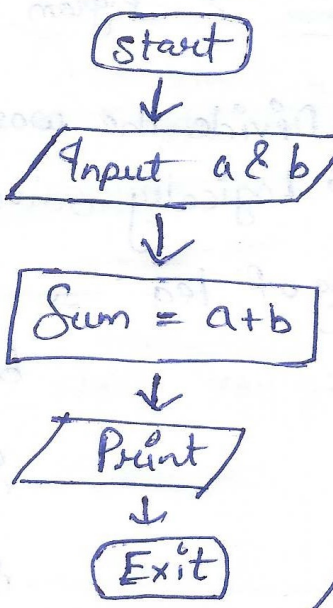
Input

First no, a

Second no., b

Output

Sum of a & b



Flowchart

- Calculate Simple intrest

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

Multiplication sign in Java → *

Output

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

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

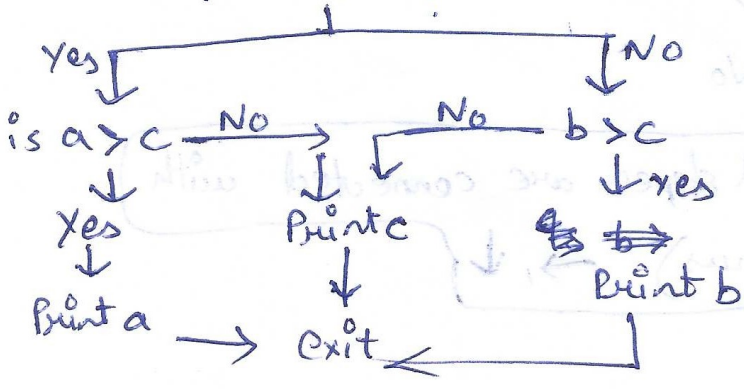
Pseudo Code

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

- Find max of 3 numbers ?

See what your brain think

⇒ 1) Input a, b, c
2) Is a > b ?



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 div → let div = 2 <n do
4) if n % div == 0 do
   Print "Not prime"
   Exit
   Else div = d+1
5) Print Prime
6) Exit

Non prime Numbers

↳ Can be divide by the no. bet n

2 ⟷ (n-1)

↳ ex- 6

2 ⟷ (6-1)

⇒ Six can be divided by 2,3

∴ It is not a prime

Q Sum of first n natural Numbers ?

$n*(n+1)/2$

⇒ Pseudo code

1) Start
2) Input no. n
3) let $n*(n+1)/2$ == a —OR—
4) Print a
5) exit

1) Start
2) Input n
3) Let value = 1, sum = 0
4) while value < n do
   Sum = Sum + Value
   Value = value + 1
5) Print
6) Exit

To check remainder in java

a % b = [ ] ← Is it equal to 0

Sum of 25 scores ? (Average)

⇒ 1) Sum = 0, count = 0
No 2) Enter S.
3) Sum = sum + S
4) Count = count + 1
5) If count = 25
6) Av sum/25

If in Question asked no. bet^n 9 and 100 then the no. are → 10 98

Solution can be found by formulae are human use.

Variables and Data Types

~~Class~~

- Name of our file and public class should always be same -

ex - JavaBacis.java

Public class JavaBascics

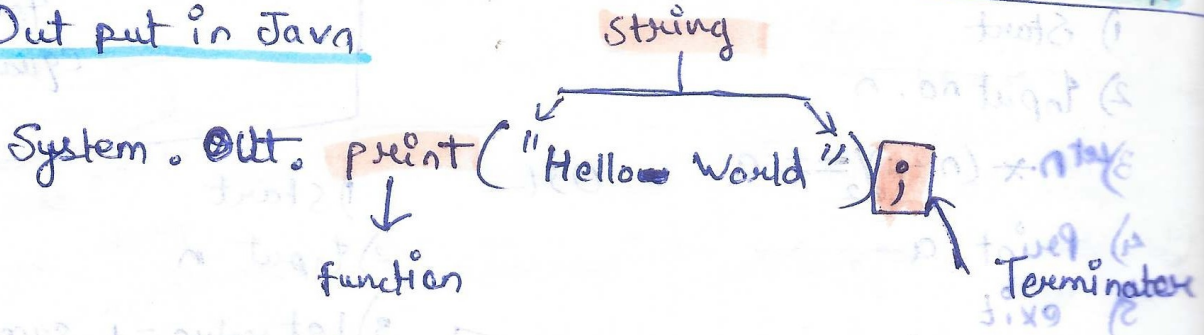
#bCompiler finds the main program first

- 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 → S should be capital

! Out → o should be small

! print → p should be small

\n → next line

or

{"Hello\n"}

Print Pattern

```
****   <- output
***
**
*
```

⇒

```
System.out.println("****");   <- Input
Syso println("***");
Syso println("**");
Syso println("*");
```

Output in Java

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

print → function

; → statement terminator.

Variables in Java

2 * (a+b)

2 → Literal

a, b → variables

- Literal → which have fixed values.
  ex - 2, 3 Character a, b, c etc.
- Variables → which can change
  ↳ 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 ⇒ 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 charecter can be stored

3) boolean

Size → 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 → A number of integer

8 byte

7) double → A long number of ~~decimal~~ a decimal number

size - 8 byte

8) ~~short~~ short

Size → 2 byte

↳ 1 number can be stored

ex- `short n = 240;`

Question - Sum of a & b

⇒
```
int a = 10;
int b = 5;
int sum = a+b
```

- Keep the name of varible Logical

Comments in Java

↳ Statements that does not execute in java

1) //it will be ignored } Single line comment
↳ Compiler ignores.

2) /* — — — */ } Multi line Comment

Input in Java

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

1) Make a class Scanner

• util is a package

```
String input = sc.next()
```

input ↑ variable

→ 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 are

→ nextByte → For byte type data
→ nextDouble — —
→ nextBoolean —
→ nextShort —
→ nextLong

Sum of a and b (By taking input)

Question

Scanner →

Program in PC

```
int a= sc.nextInt();
int b = sc.nextInt();
int sum= a+b;
Syso(sum);
```

Question

Multiplication of a and b

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

Program in PC

Question

Area of circle

Scanner

```
int r= sc.nextInt();
float area = 3.14*r*r
Syso(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 bet^n compatible types

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

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

Type Casting / Narrowing Conversion / explict → Force full type conversion

```
float a = 25.99
int b = (int) a;
Syso(b)
```

output
25

Character can also be converted into number

a = 97
b = 98
c = 99
d = 100

Type Promotion in expression

↳ Java Automatical 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

ex-

```
char a = 'a';
char b = 'b';
Syso ((int)(a));
Syso ((int)(b));
System.out.println = (b-a);
```

output
97
=> 98
1

- During expression Automatic conversion by java

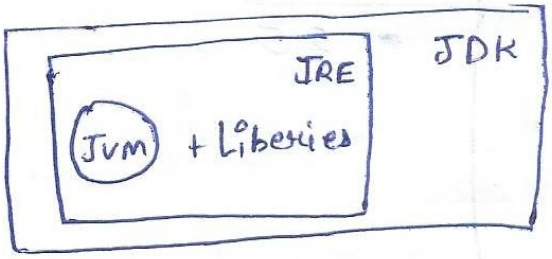
int, long, float, double → Double

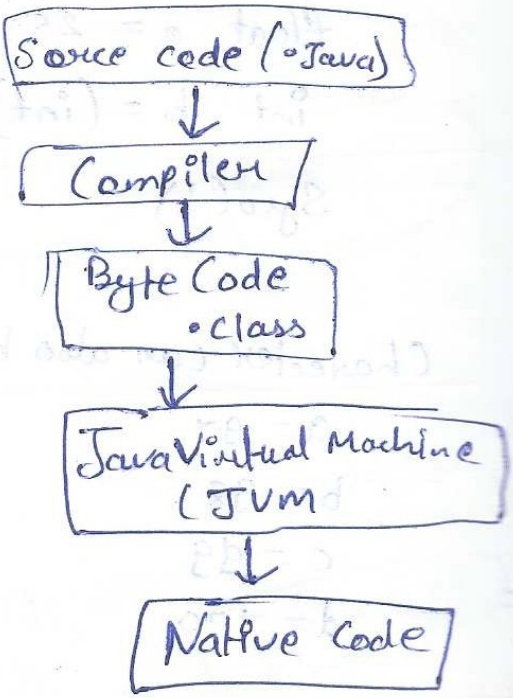
Its get converted into the greatest one

ex- int, long, float → float

How is our code is running ?

JRE → Java runtime environment





- Java is a portable language

C++ is not

- Identifiers can start with any variable including (_) and ($).

Operators in Java

add → +
Subtract → -
Multiply → *
divide → /

Sum = a+b
operands
operater
expression

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

- a = a+1 → a++ or ++a (Increment)
- a = a-1 → a-- or --a (Decrement)

#++a → Pre Increment
1) First value changes
2) then it is used

#a++ → Post increament
1) First it is used
2) then value is increased.

---

2) Relational Operators

- == To check values are equal or not
  ex- a=b → False
  (5) (10)
  We get reasult as True and False
- != not equal to
- > Smaller than
- < Greater than
- >= less than equal to
- <= greater than equal to

3) Logical Operators

True → T
False → F

i) Logical AND ⇒ &&

True && True → T
F && F → F
T && F → F
F && T → T

ii) Logical OR ⇒ ||

T || F → T
F || T → T
F || F → F
T || T = T

iii) Logical NOT ⇒ !

!T → F
!F → T

---

4) Assignment Operator

* This are faster than normal operato

i) = ex - A = B
ii) += ex - A = A + 10 → A += 10
iii) -= ex - A = A - 10 → A -= 10
iv) *= ex - A = A * 2 → A *= 2
v) /= ex = A = A/2 → A /= 2

Operator Precedence → which will execute first.

ex- int x = 3*4-1

⇒ Here x = 11
not 9.

1) [++] > [--]

2) [++] > [--] > [+] > [=] > [~] > [!]

Left to right

3) [*] > [/] > [%]

4) [+] > [-]

5) [<<] > [>>] > [>>>]

6) [<] > [<=] > [>=] > [instance of]

7) [==] > [!=]

8) [>>>=] > [>>=] > [<<=] > [|=] > [^=] > [&=] > [%=] > [/=] > [*=] > [-=] > [+=] > [=]

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

Conditional Statements

1) if-else Statements. ⇒ if(condition){

}

else{

}

i] Program → Print largest ~~2~~ numbes.

ii) Program → Print number is odd or even.

---

2) else if Statements.

⇒ if (condition 1) {

}

else if (condition2) {

}

else {

}

- Parenthesis → ()
- Curly braces → { }

i) Program → Print largest of ~~3~~ three number.

Ternary Operator

↳ 3 operands.

variable = condition ? statement1 : Statement 2;

↓ true

Nahi toh

i) if this is true the ~~s~~ variable = 1 or variable = 2

ex- boolean larger = (5>3) ? 5 : 3;

Output ⇒ 5

ex- String type = (5%2==0) ? "even" : "odd";

↳ Program → student will pass or fail?

Switch Statement

↳ If variable is 1 case one will execute.

↳ If no case is true default execute.

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

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 • Calculator

Leap year or not?

⇒ If the year is divisible by -

4 and 100 → Leap year

4 and 400 → Leap year.

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 someting
}
```

ex-

```
for(int i=1; i<=10; i++)
{ System.out.("Hellow 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

1) To take out last digit → Num % 10
2) to remove last digit → Num / 10

Take out the reverse of given number.

int rev = 0

⇒ while (a > 0) {

last digit = (a % 10);

rev = (rev × 10) + last digit

---

do-while loop

```
do {
// do something
} while(condition);
```

Break Statement → break;

↳ to get out from a loop.

Q. Break statement when user enters multiple of 10?

* If you need to keep the loop running add condition are true.

Continue Statement → continue;

↳ To skip an iteration.

Q. Display all number but not multiple of 10?

2. 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.

#1 Nested Loops

1) Print star pattern ?

```
x
x x
x x x
x x x x
```

⇒ (1) lines (4)
∴ outer loop → 4 times

(2) Number of times
∴ inner loop
If we i^{th} line
then the its i times.

(3) What to print?
⇒ "*"

```
⇒ for(int line=1; line<=4; line++){
    for(int star=1; star<=line; star++){
        Syso("*");
    }
}
```

---

2) Inverted star pattern ?

```
⇒ x x x x
  x x x
  x x
  x
```

```
⇒ for(int line=1; line<=4; line++)
    {for(int star=1; star<=(n-line+1); star++)
        { Syso("*"); }}
```

---

3) Half pyramid pattern ?

```
1
1 2
1 2 3
1 2 3 4
```

* Character is initialized using single quation.

ex- char = 'A';

Functions

↳ A block of code

(reserved name of function)

ex- Public static void main(String args[])

void → empty

main → name

(String args[]) → argument

String → array

⇒ returnType() { // body return statement; }

returnType → name

* making a new function to print hello world!

⇒ public static void printHelloworld() { System.out.println("Hello"); }

void → return type

printHelloworld → name

• 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 → Funtion that we write inside the class.

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.

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

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);
        Syso(sum);
```

Arguments

Formal parameters (Parameters)
↳ During defination of function

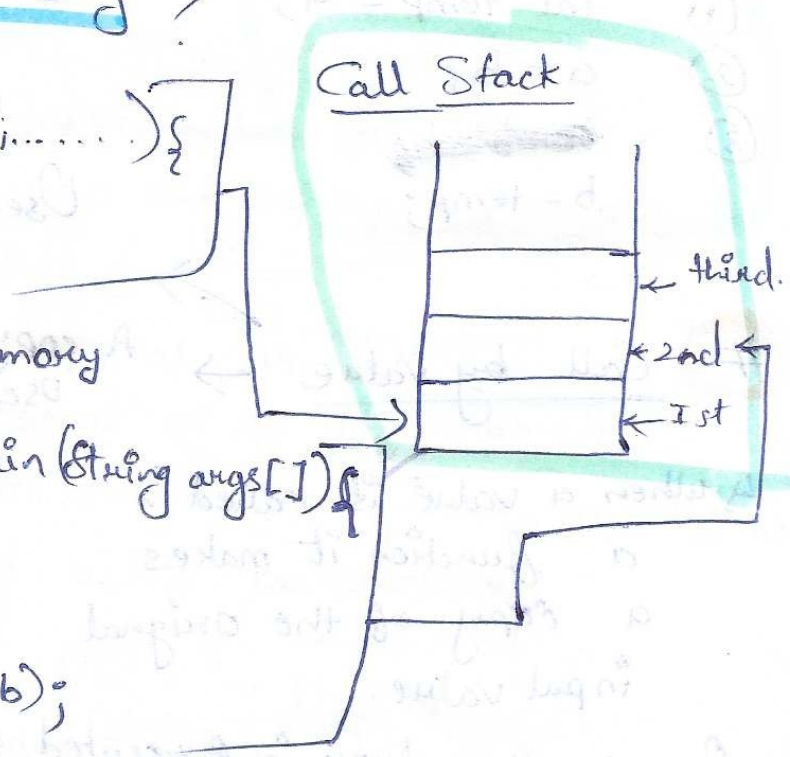
Actual parameters (Arguments)
↳ During Calling of function.

What happens in memory ?

```
Public static int sum1(i.....){
    int ...
    return sum;
}
```

It will occupies the memory

```
Public static void main (String args[]){
    Scanner -
    int a -
    int b -
    sum = Sum1(a,b);
}
```



It will occupies the memory over the first

As the program executes in main function memory
↳ Values of int a and b is saved

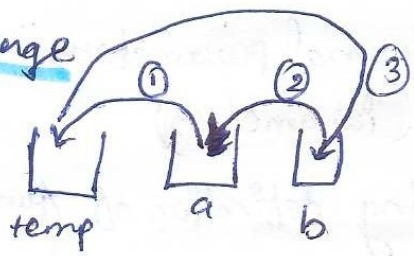
In function sum 1
↳ Sum of a and b is stored

After return function.
↳ the value is returned to main function.

Now the sum1 function memory will be deleted (empty)

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) [redacted]
    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 Refference → 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 coffeciant $^{n}C_{r} = \frac{n!}{r!(n-r)!}$

For using squareroot Use Math Class

⇒ Math.sqrt()

Inbuilt methods vs. User defined methods.

↳ Already made

ex→ sc.nextInt()

↳ Manually written

Function Overloading

↳ We can have multiple ~~multi~~ functions with same name and with different parameters

↳ Difference in that functions

- type of parameters.
- ~~type~~ 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<√n; i++){
        if(n%i==0){
            return false; }
        else { return true; }
    }
}
```

Add 2 as an exception.

Q → Check prime cannot be optimised?

Q Print all the number that are prime in the range?

Convert from Binary to Decimal

- Decimal number system → 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Binary number system → [1, 2]

Decimal	Binary
$(0)_{10}$	$= (0)_2$
$(1)_{10}$	$= (01)_2$
$(2)_{10}$	$= (10)_2$
$(3)_{10}$	$= (11)_2$
$(4)_{10}$	$= (100)_2$
$(5)_{10}$	$= (101)_2$
$(6)_{10}$	$= (110)_2$
$(7)_{10}$	$= (111)_2$
$(8)_{10}$	$= (1000)$

B → D

ex – 101

⇒ 1 0 1

↑× ↑× ↑×

$2^2 + 2^1 + 2^0$

⇒ 4 + 0 + 1 = 5

Code → ex – 1010

int pow = 0, pow++

~~int Last digit = n%10~~

Lastdigit = 1

dec = 0

dec = dec + [2 L.D × 2^{pow}]

* Math.pow(a, b) ⇒ a^b

Q. Convert binait → Decimal?

⇒ PseudoCode

```
(Num1){
int pow = 0
int deciNum = 0
While (Num1 > 0) {
int Lastdigit = Num1 %10;
deciNum = deciNum + (Lastdigit * (Int)Math.pow(2, pow));
Num1 = Num1/10; }
pow++;
```

Convert Decimal to Binary

Ex- 7

Remainder → 111

⇒

2	7
2	3
2	1
	0

← Quotient

∴ $(7)_{10} \rightarrow (111)_2$

Ex- 8

2	8
2	4
2	2
2	1

Remainder

1000

∴ 8 → 1000

Code-

```
int pow = 0;
int binNum = 0;
while (num1 > 0) {
binNum = binNum + (binNum % 2) * (int)Math.pow(10, pow);
pow++;
num = num / 2; }
Sysо(binNum);
```

while(n > 0) {
(1) divide by 2
↳ rem
(2) bin = bin + rem × 10^pow

Q. Conver Decimal to binary?

Scope → If we made a variable then

a) Class scope { } → where where we can use that variable.

b) Method Scope → In side a function.

c) for(i=1) { i only work here }

d) Block Scope

{ it will work only here }

→ A line where var Made
↓
After that everywhere in that function.

If var made here.

Math Class

ex- $x = 2$
$y = 4$

1) `Math.min(x,y)` → It will print the minimum number
⇒ 2

2) `Math.max(x, y)`
⇒ 4

3) `Math.sqrt(y)`
⇒ $\sqrt{4}$ ⇒ 2

4) `Math.pow(x,y)`
⇒ 2^4

5) `Math.avg(x, y)`
⇒ $\frac{2+4}{2} = 3$

6) `Math.abs(a)` →
↳ If a is non negetive a is returned;
↳ If a is negetive the negation of the argument returned
↳ If +0, −0 → +0 returned.
↳ If ∞ → +∞ returned.

Advanced Patterns

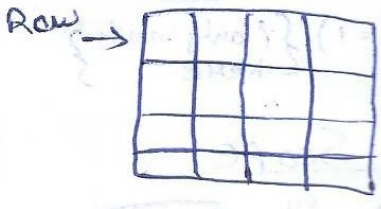
1)

```
1 2 3 4 5
*  *  *  *  *   1
*                *   2
*                *   3
*  *  *  *  *   4
```

boundry
row → 1,4
Colum → 1,5

i) total no. of lines = Rows
Loop outer (1 to 4)

Matrix

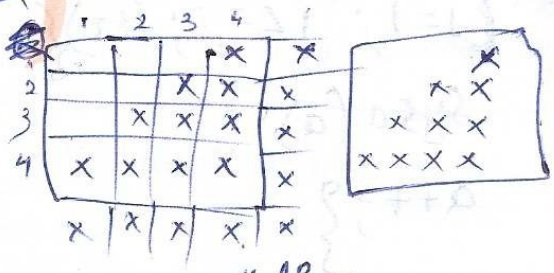


4 x 4

Print ("*") if row = 1 or 4,
Print ("*") if colum = 1 or 5,
Print (" ") if not any condition.

rectangle

Q.



```
// lines
for (i=1 ; i<=n ; i++ {

//spaces
for (j=1 ; j<=(n-i) ; j++){
    Syso (" ") ; }

// star
for ( j=1 ; j<=n ; j++){
    Syso ("*") ;}

Syso ("println") ;
```

1) First break down the Problem
2) Make a logic
3) Pseudocode
4) Program

Q.

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

lines = n = 5
to print i=1, j to 5
i=2 j to 4
1 - 3
1 - 2
1

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

Q. FLOYD'S Triangle

```
1
2  3
4  5  6
7  8  9  10
```

```
int a = 1;
for (i=1; i<=n; i++){
  for (j=1; j<=i; j++){
    Syso(a);
    a++;}}
```

Q

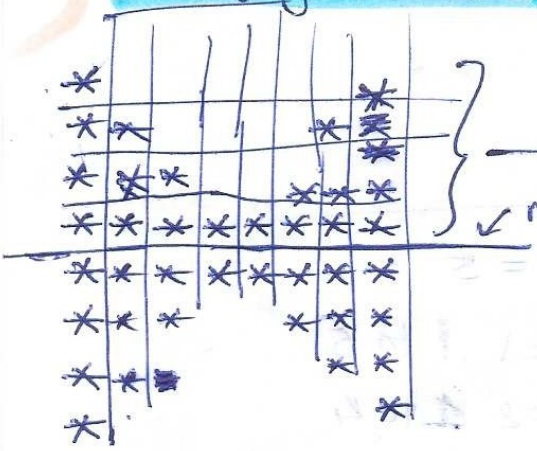
(1,1)	1			
(2,1)	0	1 (1+2)		
(3,1)	1	0	1 (3+3)	
(4,1)	0	1	0	1 (4+4)

i=1 → 1
i=2 → 0,1
i=3 → 0,1,0,1

for (i=1; i<=n;

if (i+j) = even → Syso(1)
if (i+j) = odd → Syso(0)

Q Butterfly Pattern ☆




```
for(i=1; i<=n; i++){
  for(j=1;
    for
    for
      I) n=4
        for(i=1; i<=n; i++)
        Start + space + star
```

~~if(i+j)==(n+1){~~ ① i=1<=4 ② i=1<=4

~~Syso("*");~~

~~else Syso(" ");~~

↳ for(j=1; j<=2*(n-i); j++)

	Star	Space
i=1	→	6 = 2×(n-i)
	→	4 = 2×(n-i)
	→	2 = 2×(n-i)
	→	0 =

```
* o o o o o o *
* * o o o o * *
* * * o o * * *
^ * * * * * * *
```

for first half ✓ // outer loop

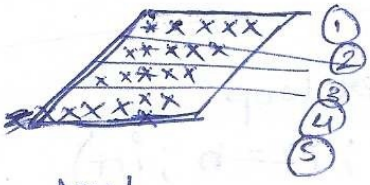
Second half → for (i=n ; i>=1 ; i--){

Syso ("*")

Inner loop same logic

For a mirror image just invert the outer loop

Solid Rhombus Pattern



	→ some space	→ star
i = 1	4	5
i = 2	3	→ 5
i = 3	2	→ 5
i = 4	1	→ 5
i = 5	0	→ 5

Outer loop

for (i=1; n times)

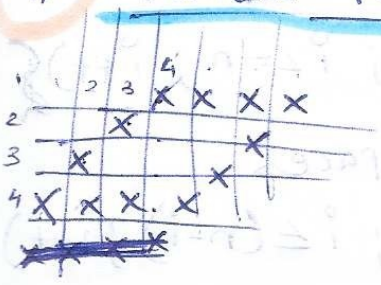
// Spaces

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

Syso (" ")

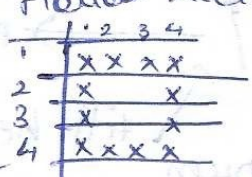
.nextline;

Hollow Rhombus Pattern



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

+ Hollow rectangle



when i==1 || i==4 and j==1 || j==4

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

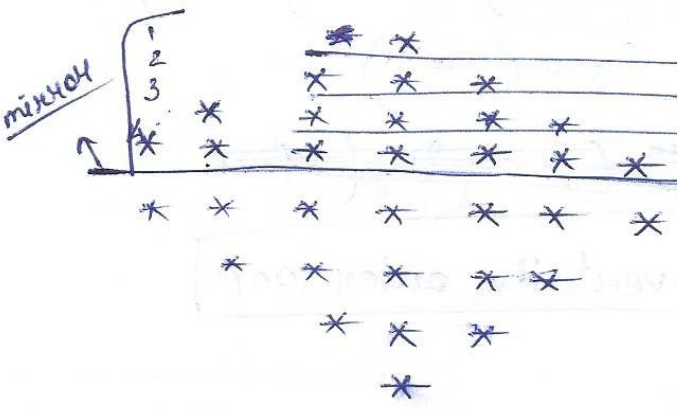
space for (j=1 ; j<=(n-i) ; j++){ syso (" "); }

Hollow rectangle

for (j=1 ; j<=n ; j++){ if (i==1 || i==n || j==1 || j==n){syso(*)}

else (syso (" ");

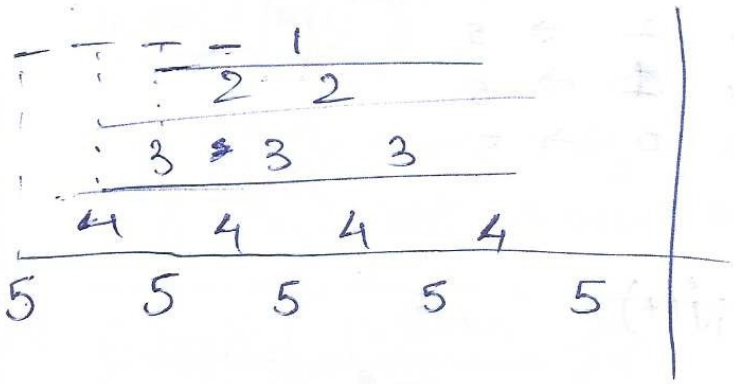
Diamond Pattern



let n=4

	Space	* star	
i=1	[n-i=4]	[i] 1	[2(i-1)+1]
i=2	[n-i]	3	2(i-1)+1
i=3	[n-3]	5	2(i-1)+1
i=4	[n-i]	7	2(i-1)+1
		add number	

Number Pyramid Pattern



outer loop

→ (i=1 ; i<=n ; i++)

→ # spaces

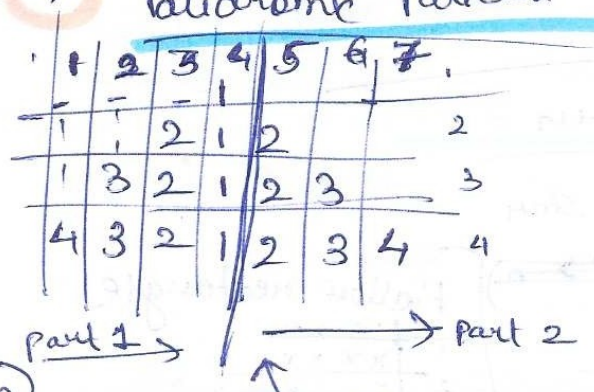
(j=1 ; j<=(n-i) ; j++)

→ # number

for(j=1 ; j<=i ; j++)

syso(i);

Palidromic Pattern



2) Part 1

num

[(j=i ; j>=1 ; j--)

syso(j)]

3) // Part 2

[(j=2 ; j<=i ; j++)

syso(j)

outer loop

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

(1) #// Spaces

(j=1 ; i<(n-i) ; i++)

*// Num

[for(j=1 ; j<=i ; j++)]

syso(j);