WHICH LANGUAGE IS BETTER ?

DEPENDS

Python

```python
# Python
import math

name = input("Your name:")
print(f"Hello, {name}")
```

C

```c
// C
#include <stdio.h>

int main() {
    char name[100];
    printf("Your name: ");
    scanf("%s", name);
    printf("Hello, %s\n", name);
    return 0;
}
```

Java

```java
// Java
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Your name: ");
        String name = scanner.nextLine();

        System.out.println("Hello, " + name );

        scanner.close();
    }
}
```
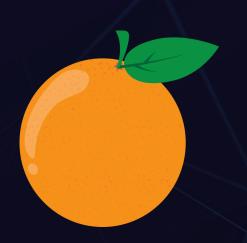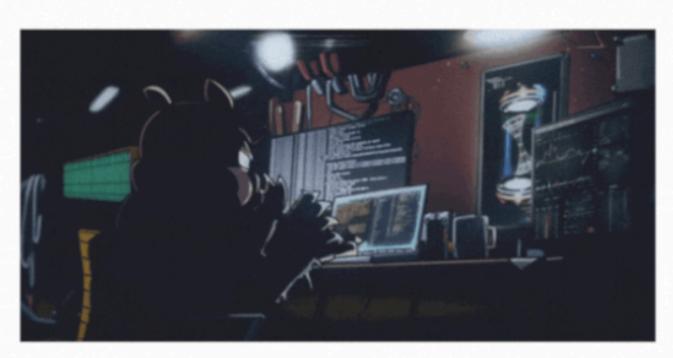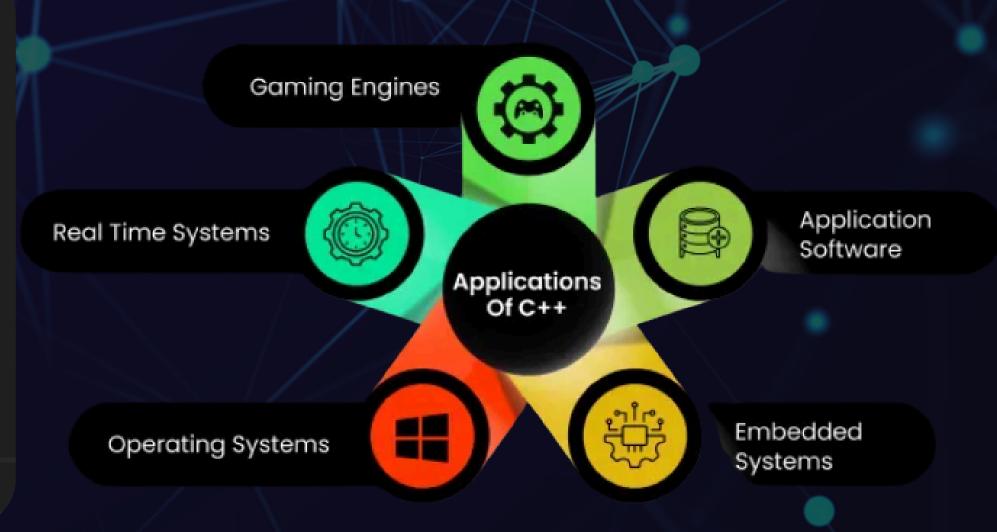
WELCOME TO
C++

SYNTAX:- SET OF RULES THAT DICTATE HOW A PIECE OF CODE
MUST BE WRITTEN
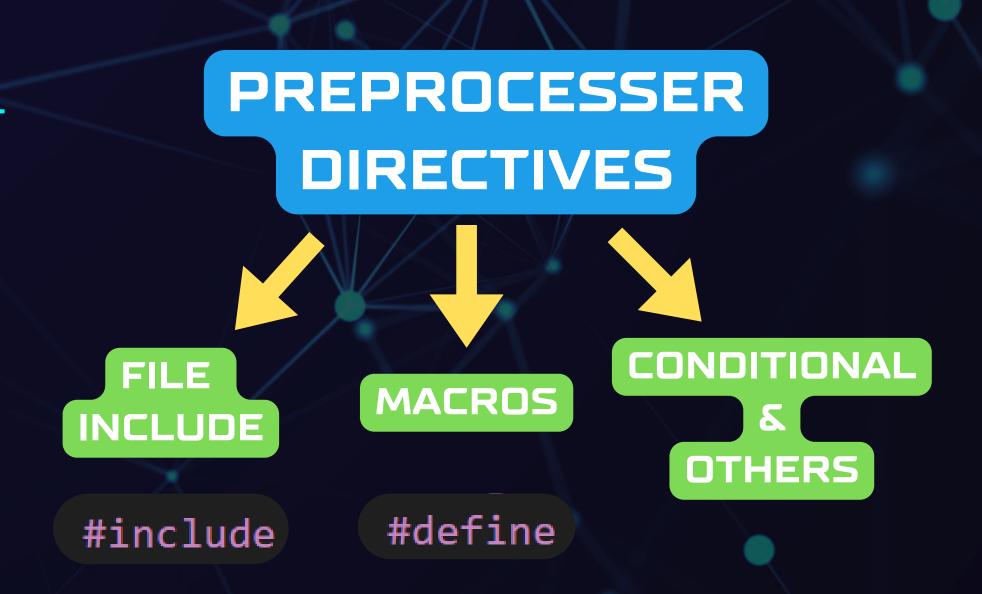
```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD 1000000007

int main()
{

    return 0;

}
```


Gaming Engines

Real Time Systems

Applications
Of C++

Application
Software

Operating Systems

Embedded
Systems

WELCOME TO
C++

```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD 1000000007

int main()
{

    return 0;

}
```

PREPROCESSER DIRECTIVES

FILE INCLUDE

MACROS

CONDITIONAL & OTHERS

#include

#define

WELCOME TO
C++

```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD 1000000007

int main()
{


    return 0;

}
```

MAIN FUNCTON

- Executes when you complie
- int main is convention
- int function has to return integer, so 0 is convention.
- After returning it will exit the function

WELCOME TO
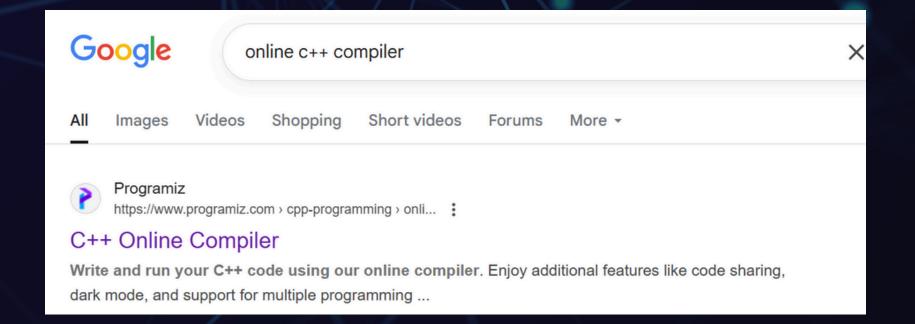# C++

```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD 1000000007

int main()
{


    return 0;

}
```


Google   online c++ compiler   ✕

All   Images   Videos   Shopping   Short videos   Forums   More ▾

Programiz
https://www.programiz.com › cpp-programming › onli...   ⋮

C++ Online Compiler

Write and run your C++ code using our online compiler. Enjoy additional features like code sharing, dark mode, and support for multiple programming ...

VARIABLES

```
int variable_name = 5;
```

VARIABLES

`int variable_name = 5;`

Data Type (dtype)

Need not memorize
Just get the feel for it
and use google to look up
and as you use, you remeber

Data Type	Size (In Bytes)	Range
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	$-(2^{63})$ to $(2^{63})-1$
unsigned long long int	8	0 to 18,446,744,073,709,551,615
signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	

VARIABLES

```
int variable_name = 5;

int Variable_name = 5;

int variable name = 5;
```

VARIABLES

```
int variable_name = 5;
```

Case sensitive

Different

```
int Variable_name = 5;
```

No Space

```
int variable name = 5;
```

VARIABLES

- The name of the variable must not contain any whitespace and special characters (ex #,$,%,*, etc).
- All the variable names must begin with a letter of the alphabet or an underscore(_).
- We cannot use C++ keyword(ex float,double,class)as a variable name

```
int variable name = 5;
```

VARIABLES

```
int variable_name = 5;
```

Operator

Value

End of statement

```
int x = 5;
int y = x + 3;
int z = y = x + 2;
```

OPERATORS

Arithmetic Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

Assignment Operators

=	Simple assignment
++	Increment
--	Decrement
+=, -=, *=, /=, %=	Combined assignment operators (e.g., x += 5 is equivalent to x = x + 5)

Relational Operators

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Bitwise Operators

&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Left shift
>>	Right shift

Logical Operators

&&	Logical AND
||	Logical OR
!	Logical NOT

OPERATORS

`int z = y = x + 2;`

SYNTAX **<variable1>** **[operator]** **<variable2>**
/constant /constant

operation(variable1, variable2)

Some change *can* happen to variables 1 and 2.
And they *can* give out a value, which can be used by other operators

Binary operators are operators that require two operands

OPERATORS

SYNTAX

<variable1> [operator]

int y = x++; **Post-Increment**

int y = ++x; **Pre-Increment**

operation(variable1)

Some change *can* happen to variables 1. And they **can** give out a value, which can be used by other operators

Unary operators are operators that only require one operand

OPERATORS

`int z = y = x + 2;`

Most operators are form
Left to Right
But assign(=), logical and
unary operators are
Right to Left

Let's Understand

There is order of operations,
but use
(parentheses[circle brackets]

```
cin>>x;

cout<<x;
```

INPUT/OUTPUT
Multiple inputs and outputs with one cin/cout

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int melons,cost;

    cout<<"Enter number of melons and cost of each melon: ";
    cin>>melons>>cost;

    cout<<"Total cost is: "<<melons<<'*'<<cost<<" = "<<melons*cost<<endl;

    return 0;
}
```

SWAP NUMBERS

Take two integers as input, and without declaring a new variable, swap the values stored in them

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int x = 5, y = 10;
    cout<<"x:"<<x<<" y:"<<y<<endl;
    //Prints 5 and 10


    //Do some magic here


    cout<<"x:"<<x<<" y:"<<y<<endl;
    //Should print 10 and 5


    return 0;
}
```

SWAP NUMBERS

Take two integers as input, and without declaring a new variable, swap the values stored in them

```cpp
#include <bits/stdc++.h>
using namespace std;


int main(){
    int x = 5, y = 10;
    cout<<"x:"<<x<<" y:"<<y<<endl;
    //Prints 5 and 10


    //Do some magic here


    cout<<"x:"<<x<<" y:"<<y<<endl;
    //Should print 10 and 5


    return 0;
}
```

```cpp
//Do some magic here

x = x + y;

y = x - y;

x = x - y;
```

HOW IT'S STORED
(& TYPE CASTING)

```
int x;

x = 65;

cout<<x;
```

**MEMORY ALLOCATED
UNINITIALISED
RANDOM VALUES**

SOME BLOCK OF
MEMORY, WHERE NOW X
IS STORED

INITIALISED

**WHEN INTERPRETED AS
INT
GIVES 65**

CHARACTERS

| Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C		
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]
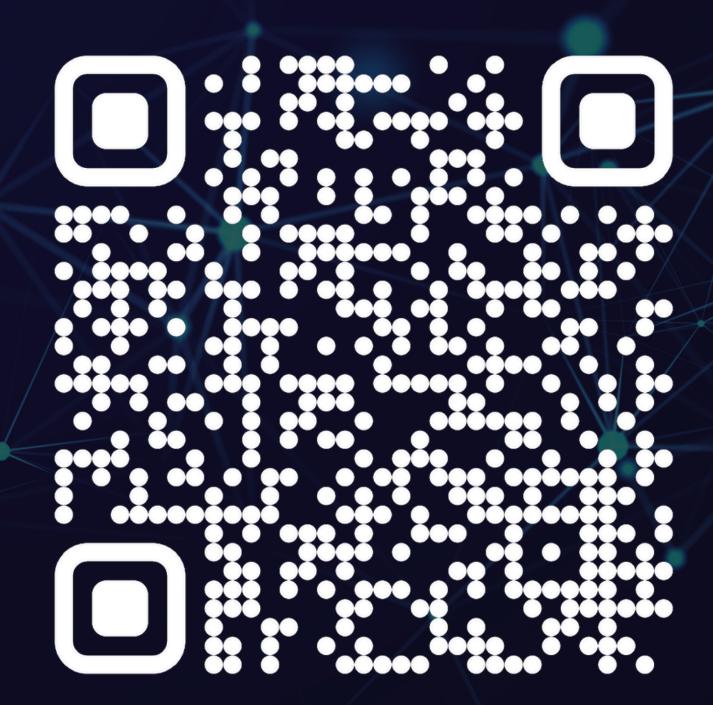
ASCII VALUES

NO NEED TO REMEMBER

FIND THE MISTAKE

5 Mistakes

Our programing club coordintator, Sathvik, has just learnt c++.
He has written a code to convert farenheit to celcius.
But the code has few errors, help him finish the code

CONDITIONALS

if else

```
if (condition1) {
    //Code for condition1
} else if (condition2) {
    //Code for condition2
} else if (condition3) {
    //Code for condition3
} else {
    //Code if no condition met
}
```

Nested if else

```
if (condition1) {
    if (condition1_1) {
        //Code 1_1
    } else {
        //Code1_2
    }
} else {
    //Code2
}
```

CONDITIONALS

Write code to take an integer input and print if the number is even or odd

CONDITIONALS

Write code to take an integer input and print if the number is even or odd

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n;cin>>n;

    if ((n%2)==0) {
        cout<<"Even"<<endl;
    } else {
        cout<<"Odd"<<endl;
    }

    return 0;
}
```

Probably looks something like this

CONDITIONALS

Write code to take an integer input and print if the number is even or odd

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n;cin>>n;

    if ((n%2)==0) {
        cout<<"Even"<<endl;
    } else {
        cout<<"Odd"<<endl;
    }

    return 0;
}
```

Single line if else

```cpp
((condition) ? (if True) : (if False));
```

```cpp
((n%2)==0)?cout<<"Even":cout<<"Odd";
cout<<(((n%2)==0)?"Even":"Odd");
```

LOOPS

For loop

```
for (initialization; condition; increment/decrement) {
    // code to be executed
}
```

Number of iterations known

While loop

```
while (condition) {
    // code to be executed
}
```

Do while loop homework

Number of iterations not known

LOOPS

Calculate factorial of a number

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n;cin>>n;
    int output = 1;
    for(int i = 1; i <= n; i++) {
        output *= i;
    }
    cout<<n<<"! is "<<output<<endl;

    return 0;

}
```

NESTED LOOPS

loop inside loop

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){

    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 3; j++) {
            cout<<'('<<i<<','<<j<<") ";
        }
        cout<<endl;
    }

    return 0;
}
```

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

NESTED LOOPS

Write a program to take an integer input n.
And print a V cone shape of '*', where the first line has total n character as shown

```
idiot@DESKTOP-0H7QRGG:~$ ./a.out
11
*         *
 *       *
  *     *
   *   *
    * *
     *
idiot@DESKTOP-0H7QRGG:~$ ./a.out
20
*                 *
 *               *
  *             *
   *           *
    *         *
     *       *
      *     *
       *   *
        * *
         **
idiot@DESKTOP-0H7QRGG:~$ ./a.out
6
*     *
 *   *
  * *
   **
```

NESTED LOOPS

Write a program to take an integer input n.
And print a V cone shape of '*', where the first line has total n character as shown


```
idiot@DESKTOP-0H7QRGG:~$ ./a.out
11
*         *
 *       *
  *     *
   *   *
    * *
     *
idiot@DESKTOP-0H7QRGG:~$ ./a.out
20
*                  *
 *                *
  *              *
   *            *
    *          *
     *        *
      *      *
       *    *
        *  *
         **
idiot@DESKTOP-0H7QRGG:~$ ./a.out
6
*     *
 *   *
  * *
   **
```

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){

    int n; cin >> n;

    int l = 0, r = n-1;
    while (l <= r) {
        for(int i = 0; i < n; i++) {
            if (i==l || i==r) {
                cout<<'*';
            } else {
                cout<<' ';
            }
        } l++; r--;
        cout<<endl;
    }

    return 0;
}
```
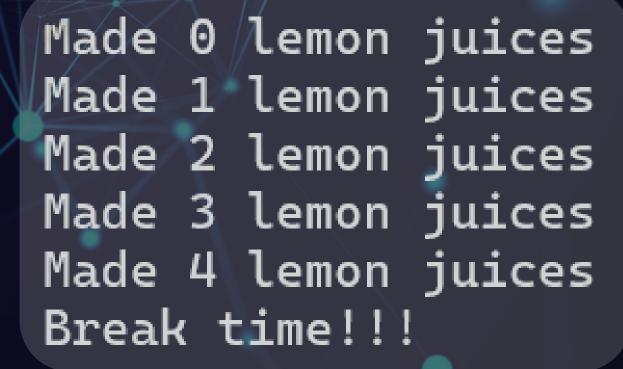
0 indexing is common notion

NESTED LOOPS

Break

Terminates the smallest enclosing loop

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){

    for (int i = 0; i < 100; i++) {
        if (i == 5) {
            cout<<"Break time!!!"<<endl;
            break;
        }
        cout<<"Made "<<i<<" lemon juices"<<endl;
    }

    return 0;
}
```

```
Made 0 lemon juices
Made 1 lemon juices
Made 2 lemon juices
Made 3 lemon juices
Made 4 lemon juices
Break time!!!
```
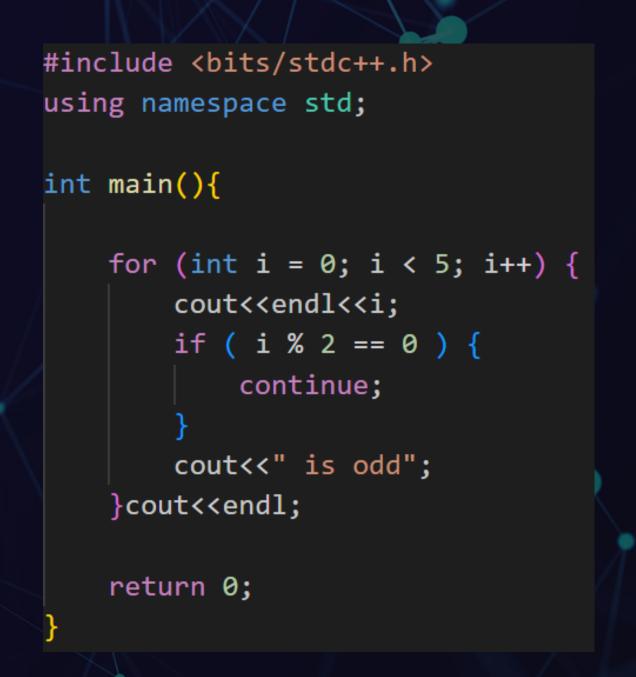
NESTED LOOPS

Continue
Skips the rest of the loop statement and starts the next iteration

```
0
1 is odd
2
3 is odd
4
```

```cpp
#include <bits/stdc++.h>
using namespace std;


int main(){

    for (int i = 0; i < 5; i++) {
        cout<<endl<<i;
        if ( i % 2 == 0 ) {
            continue;
        }
        cout<<" is odd";
}cout<<endl;

    return 0;
}
```

SCOPE

```cpp
int main(){
    {
        int i =0;
    }
    cout<<i<<endl;
    return 0;
}
```

```
test.cpp: In function 'int main()':
test.cpp:9:11: error: 'i' was not declared in this scope
    9 |     cout<<i<<endl;
      |           ^
```

Block refers to a sequence of program statements enclosed within curly braces "{ }".

Once the block is excited, varaiables declared inside are erased from memory

What happens in a block,
    stays in a block
                -Sung Jin-woo

SCOPE

```cpp
int main(){
    int i=5;
    {
        i =0;
    }
    cout<<i<<endl;
    return 0;
}
```

Gives '0' as output

Can be accesed and used by inner blocks

Global variable

Can be used anywhere

```cpp
#include <bits/stdc++.h>
using namespace std;

int i = 5;

int main(){
    cout<<"Before:"<<i<<endl;
    i = 0;
    cout<<"After:"<<i<<endl;
    return 0;
}
```

SCOPE

```cpp
int main(){
    int i=5;
    {
        i =0;
    }
    cout<<i<<endl;
    return 0;
}
```

Gives 'O' as output

Can be accesed and used by inner blocks

Global variable

Can be used anywhere

```cpp
#include <bits/stdc++.h>
using namespace std;

int i = 5;

int main(){
    cout<<"Before:"<<i<<endl;
    i = 0;
    cout<<"After:"<<i<<endl;
    return 0;
}
```

SCOPE

```
int main(){
    int x = 5;
    int x = 3;

    return 0;
}
```

It tries to re declare the same variable, but gets confused, as there in already an x.

```
int x = 5;
int main(){
    int x = 3;

    return 0;
}
```

Both are different variables with same name

SCOPE

The compiler considers the one in the innermost block(scope) which is main here

Output: 3 ←

```cpp
int x = 5;
int main(){
    int x = 3;
    cout<<x<<endl;
    return 0;
}
```

HomeWork
How to access the global x now.

CODEFORCES

Contests:
Core part of CP -
keep giving these
(loads of fun)

Problemset:
The GRIND (still
loads of fun)

Groups:
Prog Club contests!

CODEFORCES
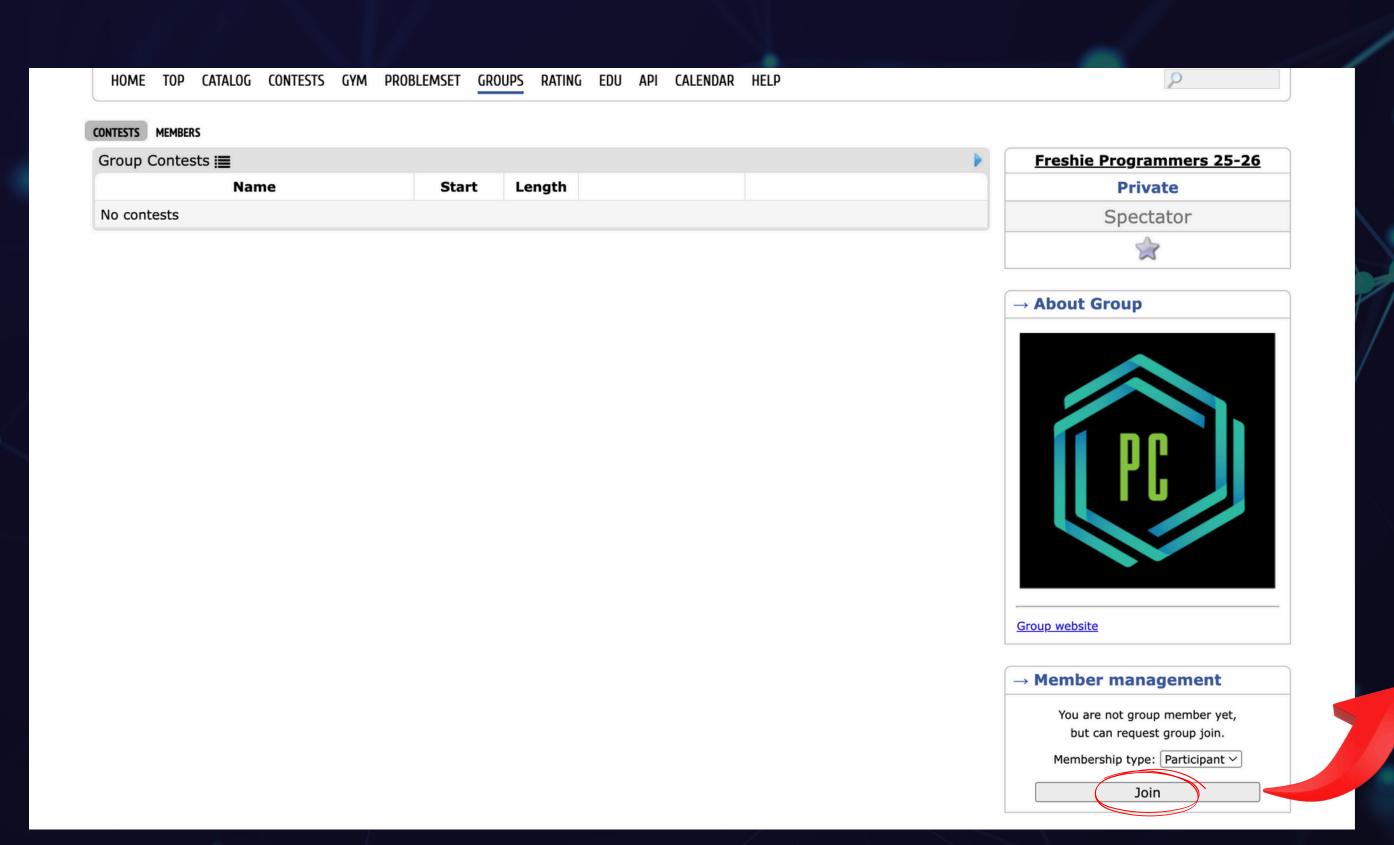
JOIN THE GROUP!

https://codeforces.com/group/M5G9QP8XsE

CODEFORCES

HOME   TOP   CATALOG   CONTESTS   GYM   PROBLEMSET   GROUPS   RATING   EDU   API   CALENDAR   HELP

CONTESTS   MEMBERS

Group Contests ☰

Name	Start	Length		
No contests				

Freshie Programmers 25-26

Private

Spectator

☆

→ **About Group**

Group website

→ **Member management**

You are not group member yet,
but can request group join.

Membership type: Participant ⌄

Join

Join here!