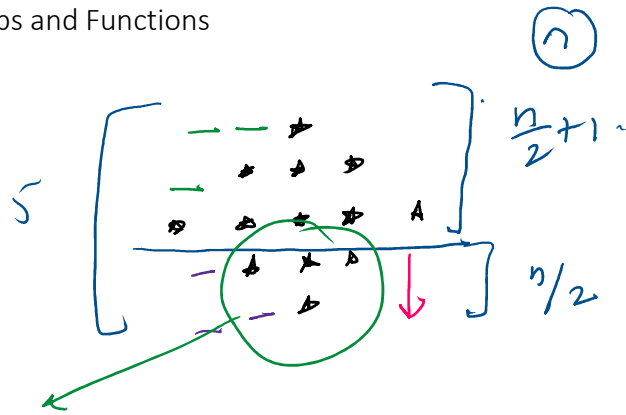


Class 4 - Nested Loops and Functions

Sunday, April 13, 2025 10:46 AM

Diamond

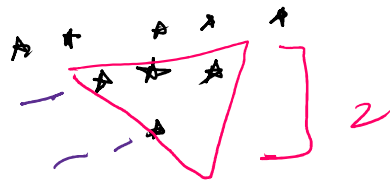


$$\frac{5}{2} + 1$$

n	i	space
2	(2)	1
2	(1)	2
		<u>n-i+1</u>

n	i	space	Stars
3	1	2	1
3	2	1	3
3	3	0	5
		<u>n-i</u>	

$2i-1$



n	i	sp
2	2	(1)
2	1	(2)
		<u>n-i+1</u>

$2i-1$

Post / Pre increment / decrement

→ Post inc/dec } Value will be updated after performing the operation.

$x++$ $x--$

→ pre. inc/dec } Value will be updated before performing the operation.

→ pre inc/dec $--x$ $++x$ → value will be -1 before perf' the operation

Some use cases of modulo and division operators

$/$ → quo

$\%$ → rem

$$7/3 \Rightarrow 2$$

$$7\%3 \Rightarrow 1$$

$$n\%3 \rightarrow [0-2]$$

$$n\%7 \rightarrow [0-(7-1)]$$

$$\begin{array}{c} 0/0/0 \\ \downarrow \\ [0-9] \end{array}$$

n	$n\%3$
0	0
1	1
2	2
3	0
4	1
5	2
6	0

$$\begin{array}{c} 0-2 \\ y \rightarrow [0-(y-1)] \\ \frac{5}{3} \textcircled{2} \end{array}$$

$$\Rightarrow [373] \text{ (10)}$$

$$\begin{array}{r} 37 \\ 10 \overline{) 373} \\ \underline{30} \\ 73 \\ \underline{70} \\ 3 \end{array}$$

③

$$\begin{array}{l} 12/10 = 2 \\ 26\%10 = 6 \end{array}$$

Ⓐ To get the last digit, take modulo with 10

⊛ To get the last digit, take modulo with 10

⊛ To get the last 2 digits, take modulo with 100

o) 373

$$\hookrightarrow 373/10 \rightarrow \boxed{37}$$

$$\hookrightarrow 373/100 \rightarrow \textcircled{3} \textcircled{7}$$

$$\begin{array}{r} 100 \overline{) 373} \\ \underline{300} \\ 73 \end{array}$$

$$12/10 \rightarrow 1$$

$$12\%10 \rightarrow 2$$

⊛ To remove last one digit, divide by 10

two ————— 100

three ————— 1000

o) Count the no. of digits

$$[\textcircled{3}\textcircled{7}\textcircled{3}] \rightarrow 4$$

$$n = 12\textcircled{3}$$

$$\left[\begin{array}{c} 123 \\ 2 \\ 12 \\ 2 \\ 1 \end{array} \right]$$

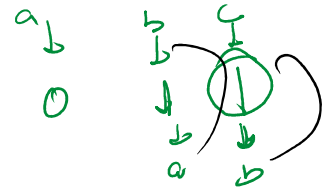
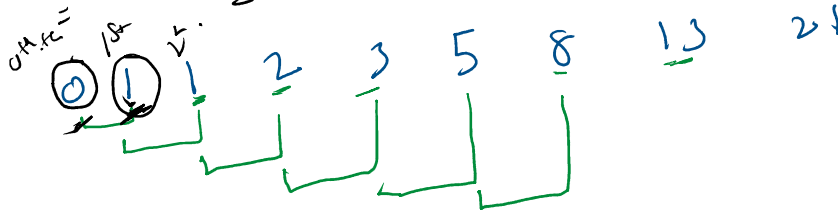
⓪

$$\begin{array}{l} \text{while } (n \neq 0) \\ \text{last dig} = n\%10 \\ \text{cout}(\text{last dig}) \\ n = n/10 \end{array} \rightarrow 3$$

$$123/10 \rightarrow 12$$

continue and break
 keywords that are used to control the flow of
 a loop.

fibonacci series



```
int a = 0;
int b = 1;
for (int i = 0; i <= n; i++) {
    cout << a;
    int c = a + b;
    a = b;
    b = c;
}
```

e

Math library

3

3¹⁰⁰⁰

Math. pow(a, b)
 \downarrow
 a^b

int a = 3 * 3 + 3

$|-3| = 3$
 $|3| = 3$

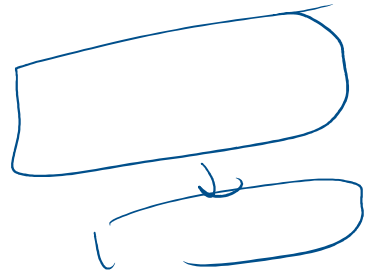
functions

$\rightarrow nCr \Rightarrow \frac{n!}{(n-r)! r!}$

$5C3 \Rightarrow \frac{5!}{2! \times 3!}$

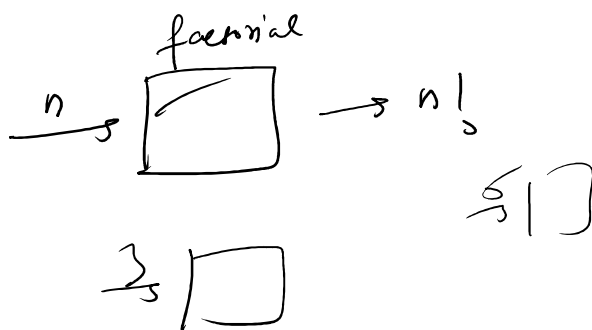
$\frac{5 \times 4 \times 3 \times 2 \times 1}{2 \times 1 \times 3 \times 2 \times 1} = 10$

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



DRY = [Don't Repeat Yourself]

function/method piece of code which will do something:



str
class
int
return fun(a, b) arguments/
inputs

return type
int fact(int n)
name
input

Ⓐ Java starts execution from
main() method.

Ⓑ functions help to reuse the code (DRY principle)

Call stack



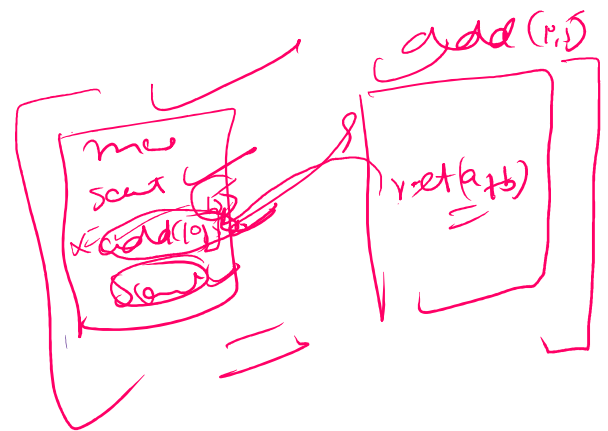
```
main() {
    x = fun(10);
    print(x);
}
```

Scope of a variable
↓
visibility

```
int add(int a, int b) {
    ret a+b;
}
```

```
main() {
    printf("Hello")
    int x = add(3,5)
    printf("Hello")
}
```

Hello
Hello



⊛ A function completes execution either when gets a return or all the lines of code written in the function are executed