Lecture 4

Output	
1.	$1^{(2^0)}$
2.	$2^{(2^1)}$
3.	~~8~~ $4^{(2^2)}$
4.	$8^{(2^3)}$
5.	$16^{(2^4)}$

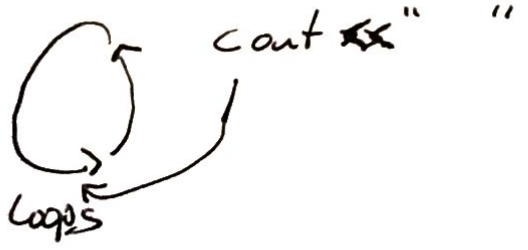
First Exam:
- Everything up to week 5

First Program
- Prompt: "Please enter positive interger:"
  Input : 5

Outputs : $2^0, 2^1, 2^2 \ldots 2^5$ or 2^n where n = input

- Can't have if/elseif, too finite and not reasonable

With Loops:



Line #	power of 2
1	1
$\xrightarrow{+1}$ 2	2
$\xrightarrow{+1}$ 3	4
$\xrightarrow{+1}$ 4	8
General → k	2^{k-1}
⋮	⋮

Keep Condition for Loop Last
- Figure out what you need the loop to do
- Then you should figure out how to control

- We are manging lineNumber and PowerOf2 in two different instances
- lineNumber is incremented by $\underline{1}$
- PowerOf2 is incremented by 2^($\underline{\text{lineNumber}}$ - 1)
- PowerOf2 is dependent on lineNumber

- An incremental approach can sometimes be easier
- But both approaches work, eith incremental or formulary.

- Can use boolean operators within condition of for loops
  exp.
```
for ( x = 1, x<y || x=y , x++ ){
    Do Something...
}
```

Program 2

- "Please enter a positive integer: "

```
4
x x x x
x x x x
x x x x
x x x x
```

line #	# of stars
1	n
2	n
3	n
⋮	⋮
$\boxed{k \to n}$	
⋮	⋮
n	→ n

← Code implements This Line

- Why do we need a nested "for" loop?
  - In this case we want the loops to be independent.
  - When n = 4
    The lineCount "for" loop keeps track of the line
    The starCount "for" loop keeps track of the stars printed
  - For each iteration of the outter loop
    the body of that iteration must be implemented

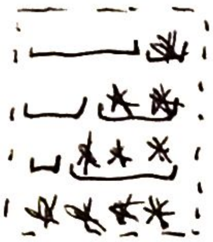
- Right Triangle

```
*
**
***
****
```

Line#	# of Stars
1	1
2	2
3	3
⋮	⋮
$\boxed{k}$	$\boxed{k}$

← Code implements This line
Line# <= # of Stars

Light Triangle Aligned to the Righ



line#	# of Spaces	# of Stars
1	n - 1	1
2	n-2	2
3	n-3	3
⋮	⋮	⋮
k	n-k	k
⋮	⋮	⋮
n	n-n = 0	n

Increment it the natural way.
- If there is No Reason to start at "0" then don't.

Program 3

Please Enter a positive integer:
2406
2406 is Super Even | 2506 ← Input
2506 is Not super Even → Output

Iterate over which direction?

$\xleftarrow{\overrightarrow{2406}}$

2406 ÷ 10 = 240 r 6
2406 mod 10 = 6
2406 div 10 = 240

↳ Better Implementation
- If you check using least sig digit you can account for number size regardless of where you are in the process

OR

2406 % 1000 = 2 r 406
2406 div 1000 = 2
2406 mod 1000 = 406

2406
2406 mod 10 = 6
6 % 2 == ∅
Super Even

2405
$\underbrace{2405 \text{ mod } 10 = 5}$
5 % 2 == 1
Super Odd = Not Super Even

4234
4234 mod 10 = 4
4 % 2 == ∅
Super Even

302514
$\neg(\text{num} > 0 \land \text{seen} = F) =$
$= (\text{num} \quad \lor \text{seen} = T)$