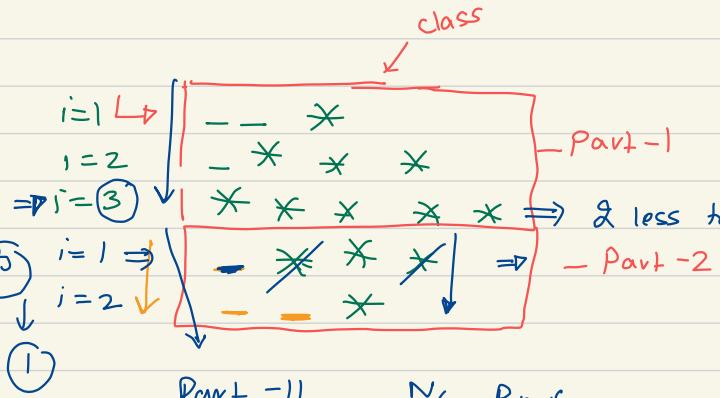
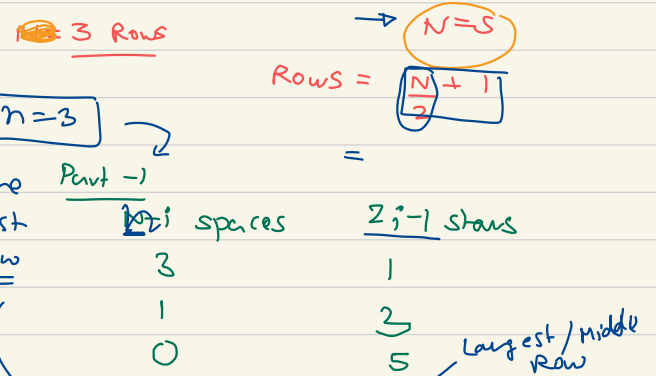
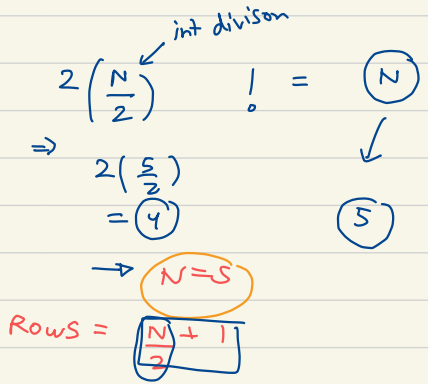
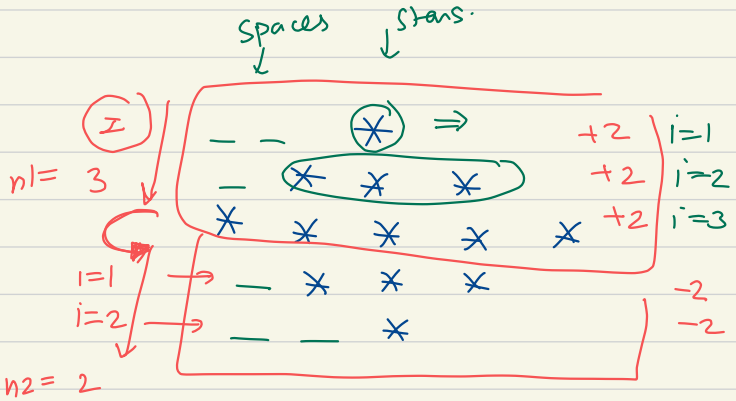
Doubts

1. ~~Diamond Pattern~~
2. ~~Integer to Char Typecast~~
3. Debugging (Class)







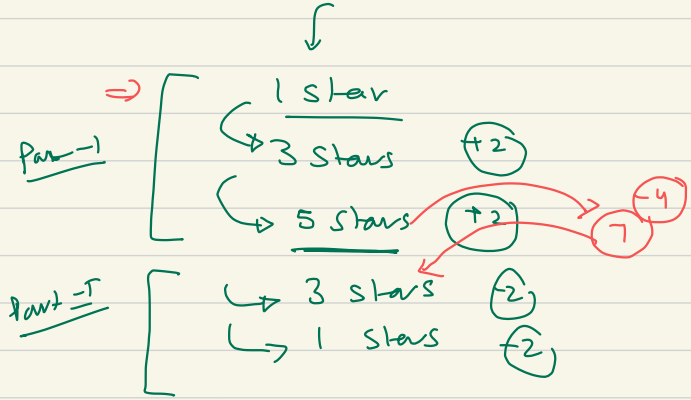


$N = 5$

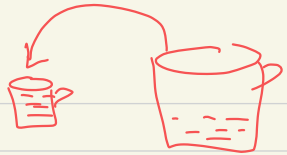
$n1 = \left(\frac{5}{2}\right) + 1 = 3$ Rows $\qquad \frac{N}{2} + 1$ Rows

$n2 = \left(\frac{5}{2}\right) = 2$ Rows $\qquad \frac{N}{2}$ Rows

i^{th} Row $= 2i - 1$ stars.



2 Int to Char



Char letter = 'A'

print(letter) : 'A'

65

char ⇒ 2 bytes

int no = 65;

65

int ⇒ 4 bytes

Char letter = (char) no;

Lossy conversion

print(no) 65;

print((char) no); 'A'

Topics Left

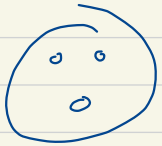
1. ++ , -- Operators — increment op
2. Ternary op
3. Switch case.

for(int i=1; i<=10, i++){

3

Post-inc i++ same as i = i+1

Pre-inc ++i same as i = i+1



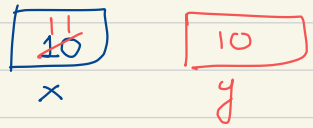
X = 10;
X++;

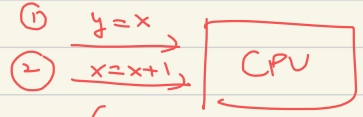
X = X+1
=> 11

Scenario : Different Behaviour

```
int x = 10;
int y = x++;
```

→ Two instructions in one line.



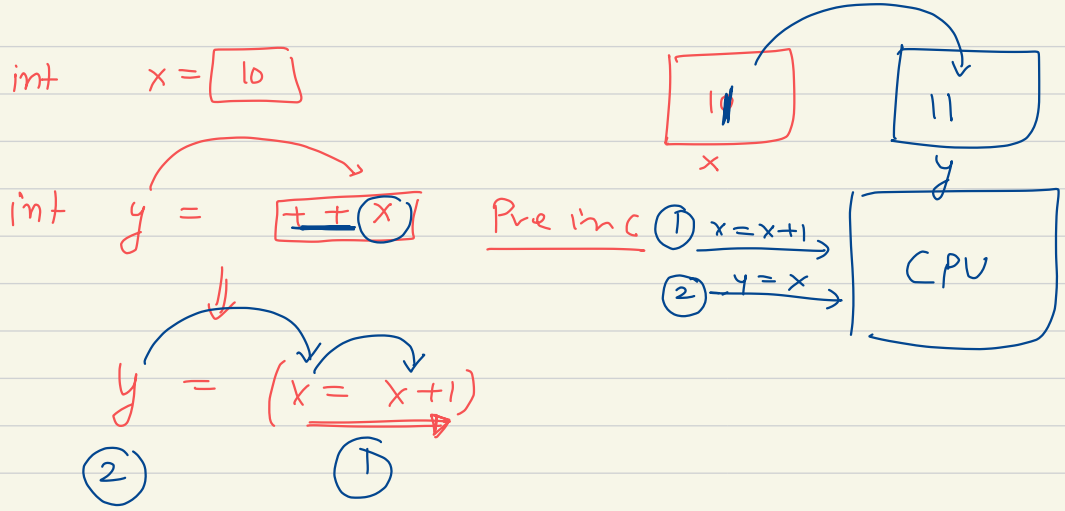


Post-increment

Assign + Increment

```
print(x, y)
```

11 10



print (x , y).

```
--x ;        // Pre-dec
x-- ;        // Post-dec
```

Compound Assignment operators

Compound Assignment

- `a = a + 5;` ⇒ `a += 5;`
- `a = a * 7` ⇒ `a *= 7;`
- `a = a % b` ⇒ `a %= b;`
- `a = a / b` ⇒ `a /= b;`

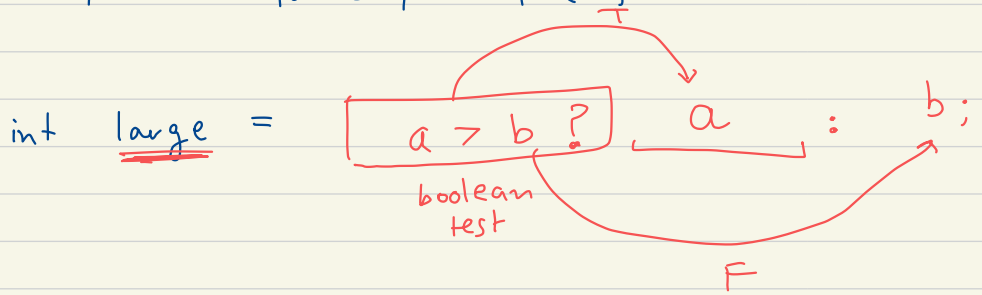
a = 6

a *= 7

⇒ (42)

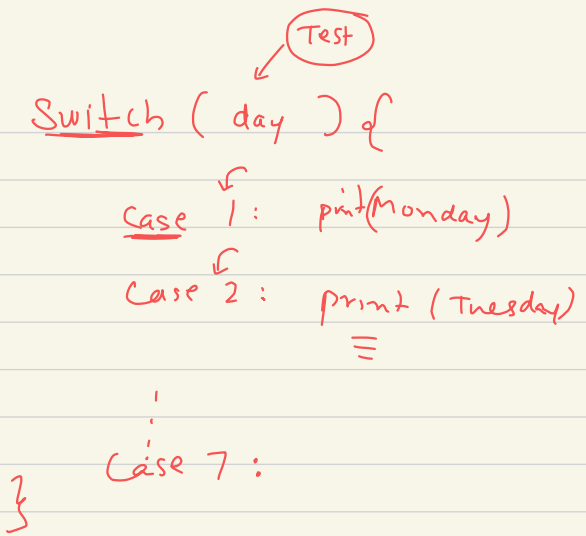
Ternary Operator ? :

↳ Replacement for simple if else,



Switch Case (another way of writing lot of if-else with simple checks)

	Input		output
if (day) =	1	-	Sun
else if (day) =	2	-	mon
	3	-	Tue
	4		
	5		
	6		
	7		


Test
Switch (day) {
case 1 : print(Monday)
case 2 : print (Tuesday)
case 7 :
}

10.30

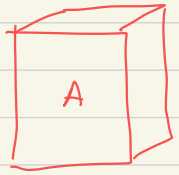
Part-2

Functions

- Debugging

Making the Code "Modular"

Solar Energy



Book 1

500 Pages

Book 2

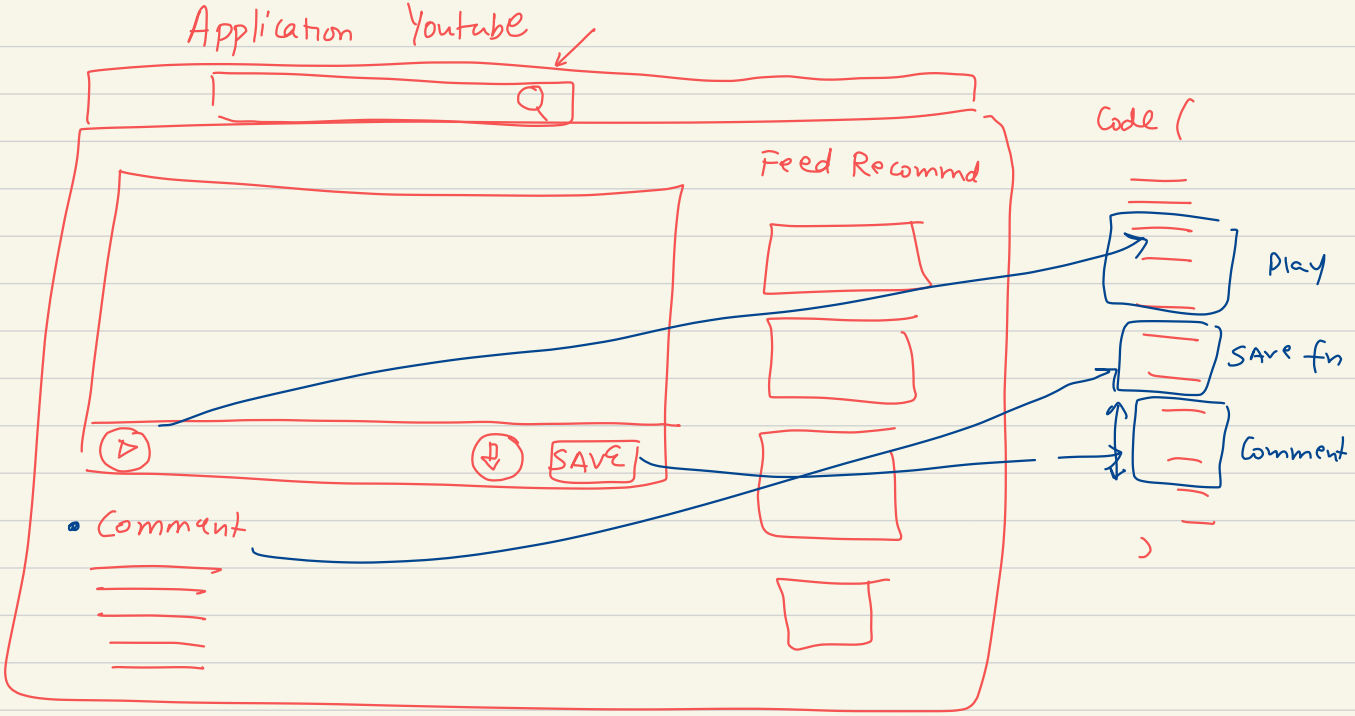
Book 2

10 chapters

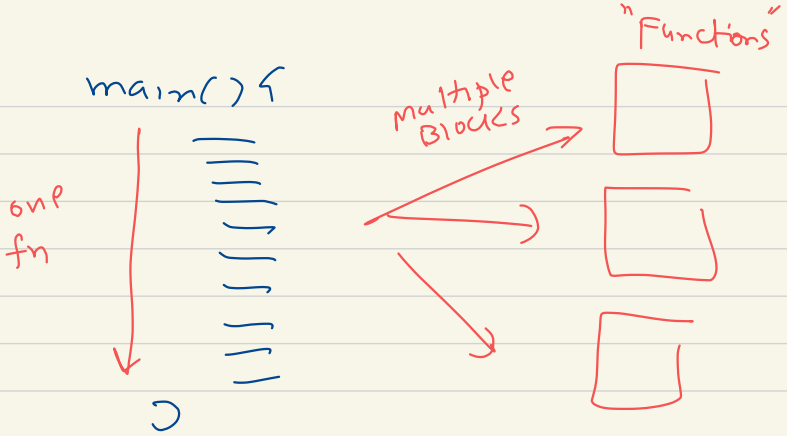
500 Pages

- Organised, Modular
- Easy to Read, Track
- Chapters can be "re-used"

```
main(){
}
```



Code → based upon what the user wants to do to Run it



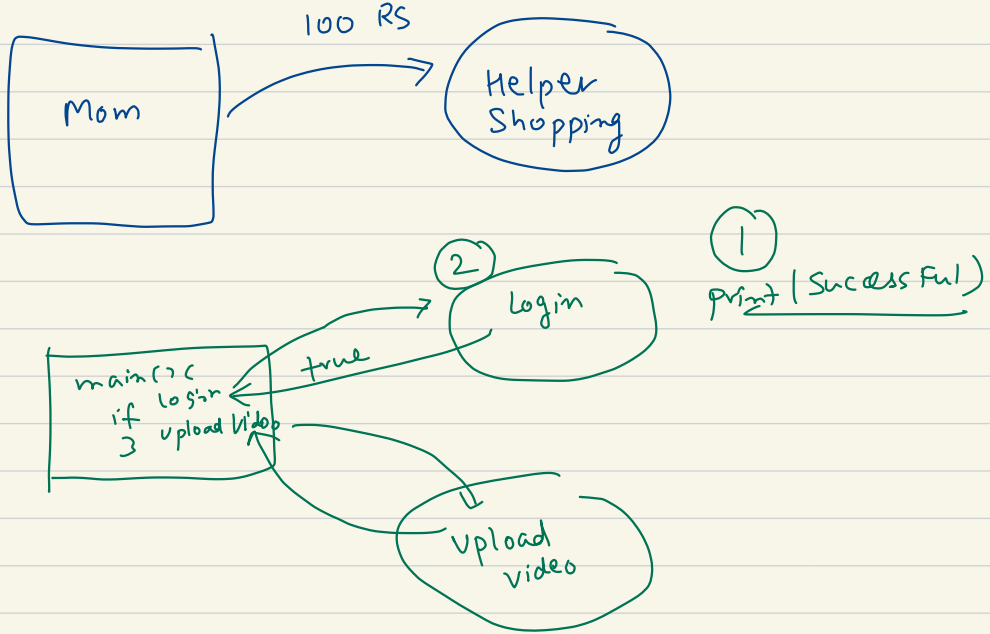
1. Defining the Fn
2. Calling (RUN) the fn

ignore for now

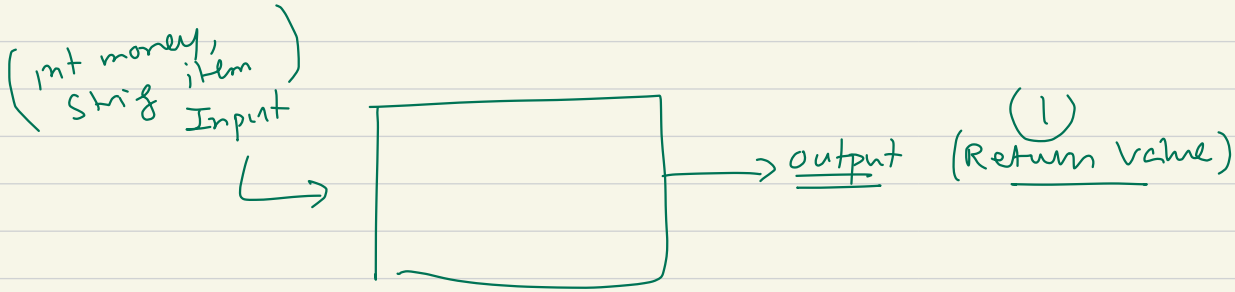
public static ReturnType fn-name (int a, int b) {

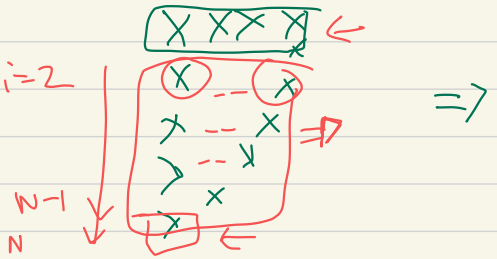
Type of output data / void => No output

Inputs


100 RS
Mom
Helper
Shopping
1
print (Successful)
2
Login
true
main.c
if login
upload Video
Upload
video

Function can accept some data (through parameters)
And they can return some data (through return values)





=>

for ( )
x
for (spaces){
}