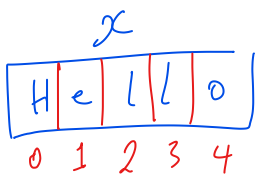
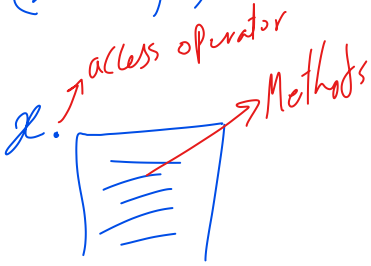
what's 7 * 4?
>> 28
Correct
what's 2 * 9?
>> 13
Wrong
what's . . . . ?
>>
_____ ?
>>
_____ ?
>>
Your score 3 out of 5

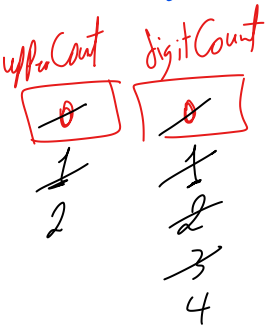
Enter num of std.: 4
Enter exam full Mark: 20
Enter std. mark: 15
$Pct = \frac{mark}{fullMark} * 100$
>=50 Pass
Fail

string x = "Hello";
cw(x) → Hello
cw(x[0]) → H
cw(x.Length) → 5





Ahmed123 X
20AhmeD20 ✓
① at least 8 char Length
② at least 2 upper isUpper(-)
③ at least 2 digits isDigit(-)



Print first 50 prime number
number: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, - - - -
all cannot divide → Prime → print
print: 2, 3, 5, 7, 11
one divide → break → not prime → no print

while _____
for _____
if _____
break

$\frac{x}{8}, \frac{y}{24}$

gcd	div
~~1~~	~~2~~
~~2~~	~~3~~
~~4~~	~~4~~
8	~~5~~
	~~6~~
	~~7~~
	8

$\frac{x}{6}, \frac{y}{9}$

gcd	div
~~1~~	~~2~~
3	~~3~~
	~~4~~
	~~5~~
	6

Call Method

Math.Min(5,2) -> 2

Math.Pow(5,2) -> 25

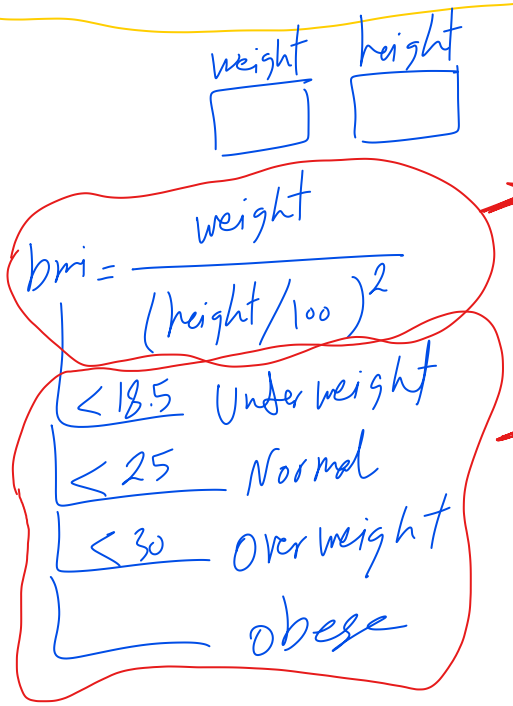
char.IsUpper('A') -> T

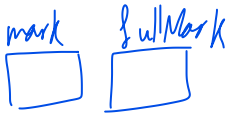
⋮

Define

modifiers return-type MethodName(parameters) {

}





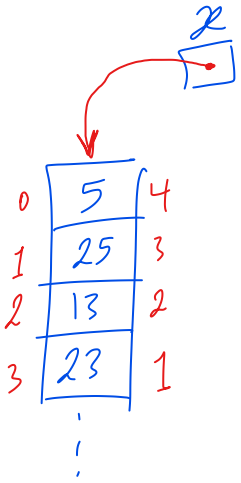
GetPercent(-,-)

$$pct = \frac{mark}{fullMark} * 100$$

GetGrade(-)

- >85 Excellent
- >75 V.Good
- >65 Good
- >=50 Pass
- Fail

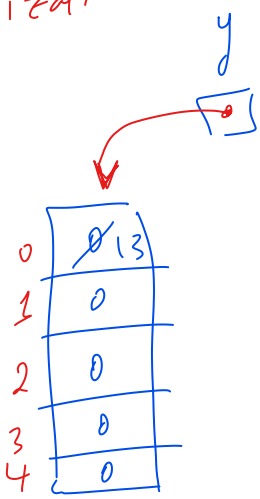
```
int[] x = {5, 25, 13, 23};
cw(x);      → Array
cw(x[0]);   → 5
cw(x[1]);   → 23
cw(x.Length); → 4
```



Declaration creation

```
int[] y = new int[5];
y[0] = 13;
```

initialization



Enter num of std.: 5

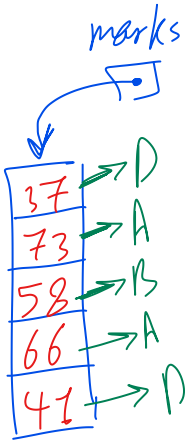
Enter std. mark: 37

~ ~ ~ : 73

~ ~ ~ : 58

~ ~ ~ : 66

~ ~ ~ : 41



A → m >= 63 best - 10

B → m >= 53 best - 20

C → m >= 43 best - 30

D → m >= 33 best - 40

F → else

best = Max()

Enter num of emps.: 4

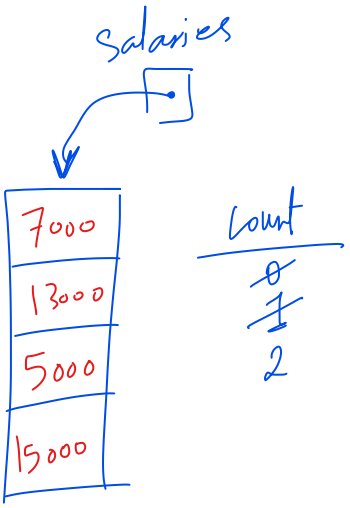
Enter emp salary: 7000

~ ~ ~ : 13000

~ ~ ~ : 5000

~ ~ ~ : 15000

avg = Average() → 10000



count

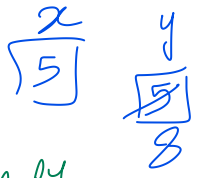
~~0~~

~~1~~

2

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

Copy value




```
int[] x = {10, 20, 30};
int[] y = x;
y[2] += 3;
```

Copy Reference