Complete Beginner While loop-2

Print all the odd numbers from 1 to 10

```
int n = 1;
while(n <= 10) {
    System.out.println(n);
    n = n + 1;
}
```

initialisation: n = 1

Printing: 1, 2, 3, ..., 9, 10 (out)

n	condition check	loop work	update
1	✓	✓	✓ (2)
2	✓	✓	✓ (3)
3	✓	✓	✓ (4)
⋮			
9	✓	✓	✓ (10)
10	✓	✓	✓ (11)
11	false		

```
int n = 1;
while(n <= 10) {
    System.out.println(n);
    n = n + 2;
}
```

n = 1

Printing: 1, 3, 5, 7, 9

n	condition check	loop work	update
1	✓	✓	✓ (3)
3	✓	✓	✓ (5)
5	✓	✓	✓ (7)
7	✓	✓	✓ (9)
9	✓	✓	✓ (11)
11	11 <= 10		

```
int n = 1;
while(n <= 10) {
    System.out.println(n);
    n = n * 1;
}
```

initialisation: n = 1

Printing: 1, 1, 1, ⋮

n	condition check	loop work	update
1	✓	✓	✓ (1)
1	✓	✓	✓ (1)
1	✓	✓	✓ (1)
⋮	⋮		⋮

Infinite loop

Loop will never stop

```
int n = 1;
while(n <= 10) {
    System.out.println(n);
    n = n * 2;
}
```

2×2

initialisation: n = 1

Printing: 1, 2, 4, 8

n	condition check	loop work	update
1	✓	✓	✓ (2)
2	✓	✓	✓ (4)
4	✓	✓	✓ (8)
8	✓	✓	✓ (16)
16	16 <= 10 false		

Problem: Bowl all balls of an over

Initialise the count = 1. and we have to deliver the ball for an over. , no. of bowls in over = 6

we have to start from count 1 & move until count is not equal to 6.

count = count+1

count	condition (count < 6)	deliver the bowl	value of count after update
1	✓	✓ → ✓	✓ (2)
2	✓	✓ → ✓	✓ (3)
3	✓	✓ → ✓	✓ (4)
4	✓	✓ → ✓	✓ (5)
5	✓	✓ → ✓	✓ (6)
6	[count < 6 False]		

make some change in condition

5 deliver

count = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~

7

```
count = 1;
while ( count <= 6) {
    // deliver a bowl
    count = count+1;
}
```

Print all perfect square from 1 to N

Ex N = 100, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

```
i = 1
while(i <= 10) {
    SOP(i * i);
    i++;
}
```

Printing	i = 1 value of i	condition i<=10	print(i*i)	update i++
1	1	✓	✓	✓ (2)
4	2	✓	✓	✓ (3)
9	3	✓	✓	✓ (4)
16	4	✓	✓	✓ (5)
⋮	5	✓	⋮	⋮
81	9	✓	✓	✓ (10)
100	10	✓	✓	✓ (11)
	11	False.		

```
i = 1
while(i * i <= 10) {
    SOP(i);
    i++;
}
```

i*i

	initialisation i=1	condition	loop work	update
1 → 1	1	✓	✓	✓ (2)
2 → 4	2	✓	✓	✓ (3)
3 → 9	3	✓	✓	✓ (4)
	4	4*4 <=10 False		

```
i = 1
while(i <= 10) {
  P1→ i = i * i;
  P2→ SOP(i);
    i++;  → update
}
```

Printing	initialisation i=1 value of i	condition	loop work Part 1 i = i*i	Part 2 SOP(i)	update
1	1	✓	1	✓	2
4	2	✓	4	✓	5
25	5	✓	25	✓	26
	26	26<=10 False			

```
i = 0
while(i <= 10) {
    SOP(i);
    i = i * i;
}
```

Syntactically

initialisation	condition	loop work	update.
i=0	✓	✓	✓(0)
	✓	✓	✓(0)

ivalue

Printing. { 0, 0 }

0
0
0
0
⋮

0

Loop will never stop.

Print all perfect square from 1 to N

Ex N=100, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

N=100

i=1

initialisation: i=1

condition: i*i <=(N=100)

loop work: S.O.P.(i*i)

update: i=i+1

```
int i=1;
while (i*i<=N) {
    SOP(i*i);
    i=i+1;
}
```

Printing.

1*1= 1 —
2*2= 4 —
3*3= 9 —
4*4= 16 —
5*5= 25
6*6= 36
49
64

3

81

100 →

value of i	condition	loop work	update
1	✓	✓	✓(2)
2	✓	✓	✓(3)
3	✓	✓	✓(4)
4	✓	✓	✓(5)
5	✓	✓	✓(6)
6	✓	✓	✓(7)
7	✓	✓	✓(8)
8	✓	✓	✓(9)
9	✓	✓	✓(10)
10	✓	✓	✓(11)
11	121 <=100 false		

Start from 10:40 → Break

Problem: Given a number, Print the last digit of a number?

$N = 435\underbrace{8}_{\text{last digit}}$

% → modulo operator] → gives remainder.

$4358 = 4350 + 8$

$\frac{4358}{10} = \frac{4350 + 8}{10} = \frac{4350}{10} + \frac{8}{10}$

completely divisible by 10

Remainder on 10

last digit = N % 10

$l.d = 4358 \% 10 = 8$

Repeat.

$\underbrace{l.d}_{\text{last digit}} = N \% 10$

① How is it working?

② why only 10?

$N = 4358$

$l.d = (4358) \% 10 = (4350 + 8) \% 10$

$= (4350 \% 10) + (8 \% 10)$

$l.d. = 0 + 8 = 8$

$\frac{4358}{10} = \frac{4350}{10} + \frac{8}{10}$

completely div

Remainder

```
      435
10 ) 4358 (
     40
      35
      30
       58
       50
        8
```

last digit = N % 10

Problem: Given a Number, print all the digits of a number.

N = 6 3 4 1 → print → 1
4
3
6

Starting direction from least significant digit to most significant digit

N = 6 7 4 2

Sol

SOP(n % 10): → 2

SOP(n % 100) → 43

SOP (n % 1000) → 743

SOP (n % 10000) → 6743

⋮

Assume: no. of times we have iterate

same where.

N = 0 → '0' is also a digit

```
if(N==0){
    SOP(0);
}else{
    approach of print all digit
}
```

number = 7 4 3 6

```
        743
10 ) 7436 (
     70
      43
      40
       36
       30
        6 → Remind.
```

l.d = number % 10 ⇒ 6

number = number / 10

number	l.d.	num=num/10
7436	6	743
743	3	74
74	4	7
7	7	0
0	0	

→ Break point

No. of digits →

N = 34562

Count = 0

```
N = 34562
count = 0;
while(N >= 0) {
    N = N / 10;
    count++;
}
SOP(count);
```

Count	N	condition	work	Update
0	34562	✓	3456	1
1	3456	✓	345	2
2	345	✓	34	3
3	34	✓	3	4
4	3	✓	0	5
5	0	✓	0	6
6	0	✓	0	7
7	0	. . . . .		

loop will never End.

NOTE: If we want to iterate on a number, we have made a condition $n > 0$] → moving condition

Prod = 1 → 6 time.

iterate 2 time.

prod = prod * 4.

4

1

4 ← After 1st.

16 ← " 2nd "

64 ← After 3rd it

4 ^ 3 = 4 * 4 * 4 =

= 4 * 4 * 4

Print table :

Print the table for N.

N = 4.

```
int i=1;
while( i<=10 ) {
    SOP (N + "*" + i + "=" + (n*i));
    i++;
}
```

boolean Expression

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = ⋮
4 * 8 =
4 * 9 = ⋮
4 * 10 = 40

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
⋮
4 x 10 = 40.

```
if ( 1) {
    // gf statement
} else {
}
```

1	2	4	8

16 → n After value.

n	initialisation	condition n<=10	work SOP(n)	update n = n*2
1	n=1	✓	✓	✓ (2)
2		✓	✓	✓ (4)
4		✓	✓	✓ (8)
8		✓	✓	✓ (16)
16		False.		

compu