While loop-1

Loop: Same task multiple times

Example:

```
int i =1;
while (i<=5) {
  S.o.p (i);
   i = i+1;
}
```

i	i<=5	o/p	i=i+1
1	T	1	2
2	T	2	3
3	T	3	4
4	T	4	5
5	T	5	6
6	F	(it's time to get out from loop)	

(i.e) break out from loop

STRUCTURE OF WHILE LOOPS

Step1: Initialize a loop variable

```
int i =1;
```

Step 2 : write while condition

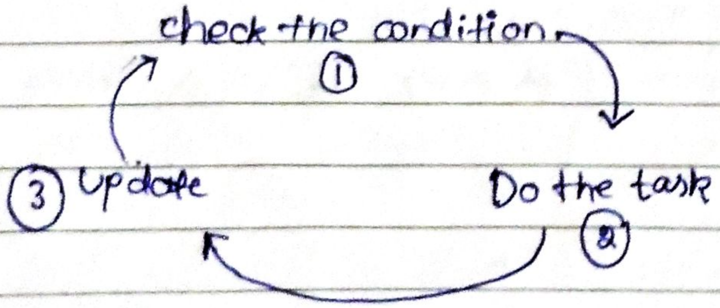
```
while (i<=5){ [till what moment while will execute]
```

Step 3 : write code about your task (printing)

```
S.o.P(i);
```

Step 4 : Updation of loop variable

```
i = i+1 ;
}
```



_/_/_

① Print all numbers from 1 to 5

```
// intialisation of loop variable (Step 1)
      int i =1;
// while with condition            (Step 2)
     while (i<=5) {
//   code for task                 (Step 3)
       System.out.println(i);
//   updation of loop variable  (Step 4)
       i= i+1;
    }
```

② print all numbers from 1 to n

```
  int n= scn. nextInt();       Custominput : 20
  int i =1;                    output : 1,2,3,.... 20
  while (i<=n) {
  S.o.p (i);
  i++;  // i=i+1
 }
```

③ Given n, print all even nos. from 0 to n

Custominput : 13 ; output : 0,2,4,6,8,10,12

custom input : 6 ; O/p : 0,2,4,6

if n=6

(i)

```
  int i= 0;
  while (i%2 ==0) {
     ---
     ---

  }
```

i	i%2==0	SOP	i=i+1
0	T	0	1
1	F		

(Wrong code) → because not using 'n' anywhere in while condition.

(ii) Correct code is,

```
intn= scn.nextInt();
int i=0;
while (i<=n) {
        if (i%2==0) {
            S.o.p (i);
   }
        i=i+1;
}
```

input : n=6

i	i<=6	i%2==0 (O/P)	i=i+1
0	T	0	1
1	T		2
2	T	2	3
3	T		4
4	T	4	5
5	T		6
6	T	6	7
7	F	(break out from loop)	

(iii) Wrong method

f n=4

```
int i=0
 while (i<=n && i%2==0) {
 S.o.p(i);
  i=i+1;
}
```

i	i<=4 && i%2==0
0	T && T
1	T && F (break...

O/P : 0

(iv) Another correct method

n=6

```
 int n= Scn.nextInt();
 int i=0;
 while (i<=n) {
  SOP(i);
  i=i+2;
}
```

i	i<=6	O/P	i=i+2
0	T	0	0+2=2
2	T	2	4
4	T	4	6
6	T	6	8
8	F	(break out...	

Previou solution/output (ii sum) → i : 0, 1, 2, 3, 4, 5 . . .

current o/p (iv) → i : 0, 2, 4, 6

④ given n, print all odd numbers from 1 to n

(i)

n=6

O/P:
1
3
5

```
int n= scn.nextInt();
int i =1;
while (i<=n) {
    if (i%2 != 0) {
        s.o.p (i);
    }
        i=i+1;
    }
```

i	i<=n	i%2!=0	i=i+1
1	T	1	2
2	T	-	3
3	T	3	4
4	T	-	5
5	T	5	6
6	T		7
7	F (breakout...		

(ii)

n=6

O/P:
1
3
5

```
int n= scn.nextInt();
int i= 1;
while (i<=n) {
   s.o.p(i);
   i = i+2;
}
```

⑤

```
int i=5
while (i<=10){
  s.o.p (i);
   i = i*2;
}
```

i	i<=10	O/P	i=i*2
5	5<=10	5	10
10	10<=10	10	20
20	20 (break out)		

⑥

```
int i=0
while (i<=10) {
  s.o.p (i);
  i= i*2;
}
```

i	i<=10	o/p	i=i*2
0	0<=0	0	0
0	0<=10	0	0
0	0<=10	0	0
⋮			

Endless (infite loop)

_/_/_

⑦ Given n, print multipies of 4 from 4to n

f n= 21 n=21 ⇒ 4 8 12 16 20

n=16 ⇒ 4 8 12 16

(i) Correct method, when n=9

```
int n= scn.nextInt();
int i=4;
while (i<=n) {
    if (i%4 ==0) {
        s.o.p(i);
    }
    i=i+1;   // i++
}
```

i	i<=9	O/p	i++
4	T	4	5
5	T		6
6	T		7
7	T		8
8	T	8	9
9	T		10
10	F	(break out--	

(ii) Another correct method

```
int n = scn.nextInt();
int i=4;
while (i<=n) {
  s.o.p(i);
    i= i+4;
}
```

i	i<=9	O/P	i=i+4
4	T	4	8
8	T	8	12
12	F	(break out.	

Comparison (i) & (ii) if n=17

(i) i → ④ 5,6,7,⑧ 9,10,11,⑫ 13,14,15,⑯ 17

(ii) i → 4 8 12 16

⑧ Print numbers from 5 to 1

```
int i=1;
while(i<=5){
  S.o.p(i);
  i=i-1;
}
```

i	i<=5	O/P	i=i-1
1	T	1	0
0	T	0	-1
-1	T	-1	-2
-2			
⋮	(infinite loop)		

__/__/__

(9)

```
int i=5
while (i>=1) {
   SOP(i);
   i--;
}
```

i	i>=1	O/P	i=5-1
5	T	5	4
4	T	4	3
3	T	3	2
2	T	2	1
1	T	1	0
0	F	(breakout	

(10) print all numbers from n to 1

n=7

```
int n= scn.nextInt();
int i=n;
while (i>=1) {
   S.O.P (i);
   1--;
}
```

O/P:

```
7
6
5
4
3
2
1
```

(11) Given a positive int A, print sum of all odd nos. in the range [1, A]

```
Scanner scn = new Scanner (System.in);
int A = scn.nextInt();
int i=1, sum =0;
while(i<=A) {
      sum = i+sum;
     i = i+2;
  }
     S.O.P (sum);
 }
}
```

(12) Given +ve int A. print sum of all even nos in the range [1, A]

```
int A = scn.nextInt()
int i=2, sum=0;
while (i<=A) {
      sum= i+sum;
      i=i+2;
  }
  S.O.P(sum);
}
}
```