Date 03/10/2023
Page

while loop

while loop is similar to for loop. i.e while loop is alternative to for loop, People use as per their comfort.

e.g:-

1 to n

using for loop:

initialisation, condition to break loop, updation.

```
for (i=1; i<=10; i=i+1)
{
    cout<<i<<" ";
}
```

using while loop:

syntax:-

```
initilize
while(break)
{
  update
}
```

```
int i=1;
while(i<=10)
{ cout<<i<<" ";
  i=i+1;
}
```

i=1 2 3 4 ... 10

print 1 2 3 4 10

11 <= 10 → false → out of loop

★ Wap to print table using while loop.

n = 8 →

```
int i=1;
while(i<=10)
{
  cout<<n<<"x"<<i<<"="<<n*i<<endl;
  i=i+1;
}
```

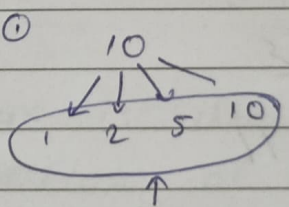
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
⋮
8 x 10 = 80

Refer code: 09-while and do-while loop in c++ / Class code / Table.cpp.

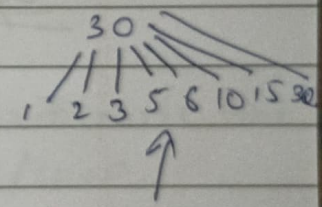
★ Wap to print factors of n.

Factors ← number that can divide n.

e.g



factors of 10.



factors of 30.

```
int i=1;
while(i<=n)
{
  if(n%i==0)
    cout<<i<<" ";
  i=i+1;
}
```

Refer code:.../factors.cpp

★ H/w odd / even → All 1 to n using while loop.

do-while loop

for / while loop	do-while loop
① initialize	① initialize
② update	② break condition
③ break condition	③ update
↑ in for/while loop	↑ in do-while loop.

In do-while loop first ~~print~~ execute the body then condition check, so even if condition is false initially then also loop will be executed once.

Syntax :-

```
do{
    //
    // update
}while (break);
```

e.g.:- print 1 to n

```
int i=1;
do{
    cout<<i<<" ";
    i++;
}while(i<=n);
```

n = 5

i=1 print 1 — 1<=5 true

i=2 print 2 — 2<=5 true

i=3

:

* WAP to print sum of n natural no.

n = 10

1 to ro

sum = 0

sum = sum+i;

i = 1

do Program :

Sum=0	
Sum=sum+i → 1	1
3	2
6	3
10	4
15	5
21	6
28	7
36	8
45	9
55	10

```
int i=1, sum=0;
do{
Sum=sum+i → Sum += i
    i++;
}while(i<=10)
cout<<sum;
```

Break, Continue

Break: Break is used to terminate the ~~loop~~ flow of program/loop.

Continue: Continue is used to skip the ~~current~~ current execution and continue with next itiration.

e.g.: Suppose we want a program that will print 1 to n, but when i==5, ~~loop~~ we don't want further to print anything.

```
i=1
while(i<=n)
{ if(i==5)
    break;
  cout<<i;
  i++;
}
```

op: 1 2 3 4

e.g Print 1 to n
if x no. then don't print further.

```
n=10
x=7
i=1
while(i<=n)
{
  if(i==x)
  {
    break;
  }
  cout<<i;
  i++;
}
```

continue:-

Skip current itiration.

e.g: Wap to print 1 to n, if n divide by 4 then skip the number.

```
~~i=1~~
~~for~~ for(i=1; i<=10; i++)
  { if(i%4==0)
      continue;
    cout<<i<<" ";
  }
```

op: 1 2 3 5 6 7 9 10

Switch

```
i = 1 -> Rohit
i = 2 -> Mohit
else  -> Sohit
```

->

```
if(i==1)
    cout<<"Rohit";
else if(i==2)
    cout<<"Mohit";
else
    cout<<"Sohit";
```

Above program can also be written using switch statement.

~~i==1~~

Syntax.

```
i=1
switch( i )        // i: variable
{
  case 1:
      // break;
  case 2;
      // break;
  :
  default:
      // break;
}
```

```
switch(i)
{
  case 1;
      cout<<"Rohit";
      break;
  case 2;
      cout<<"Mohit";
      break;
  default:
      cout<<"Sohit"
      break;
}
```

if we don't use break in case then "fall through" will happen. i.e all the ~~statement~~ below case will also be executed

Refer code : .../switch.cpp.

e.g:-

```
char name = ' ';
switch (name)
{
  case 'a':
        cout << "Hello";
        break;
  case 'b':
        cout << "Nope";
        break;
}
```

* default in switch is optional.
Refer code : ../switch.cpp

on case we can use int, char but can't use float or double type of value.

Scope of a variable

- Block start from '{' and end at '}'.
- scope of a variable is within block only (curly braces).

- within a same block duplicate variable name is not valid. but in case of child block duplicate variable name is valid.
child block ← block within block.

e.g (i)

```
int main()
{
  int num=10
  cout<<num; //10
  int num=20; X not valid.
}
```

(ii)

```
int main()
{
  int num=10
  cout<<num; //10
  if(n==10)
  {
    int num=20; ✓ valid.
    cout<<num; // 20
  }
  cout<<num; //10.
}
```

Here num's scope is within if block only ouside if block this num is not available.

O/P: 10 20 10.

Refer code: ../scope.cpp.