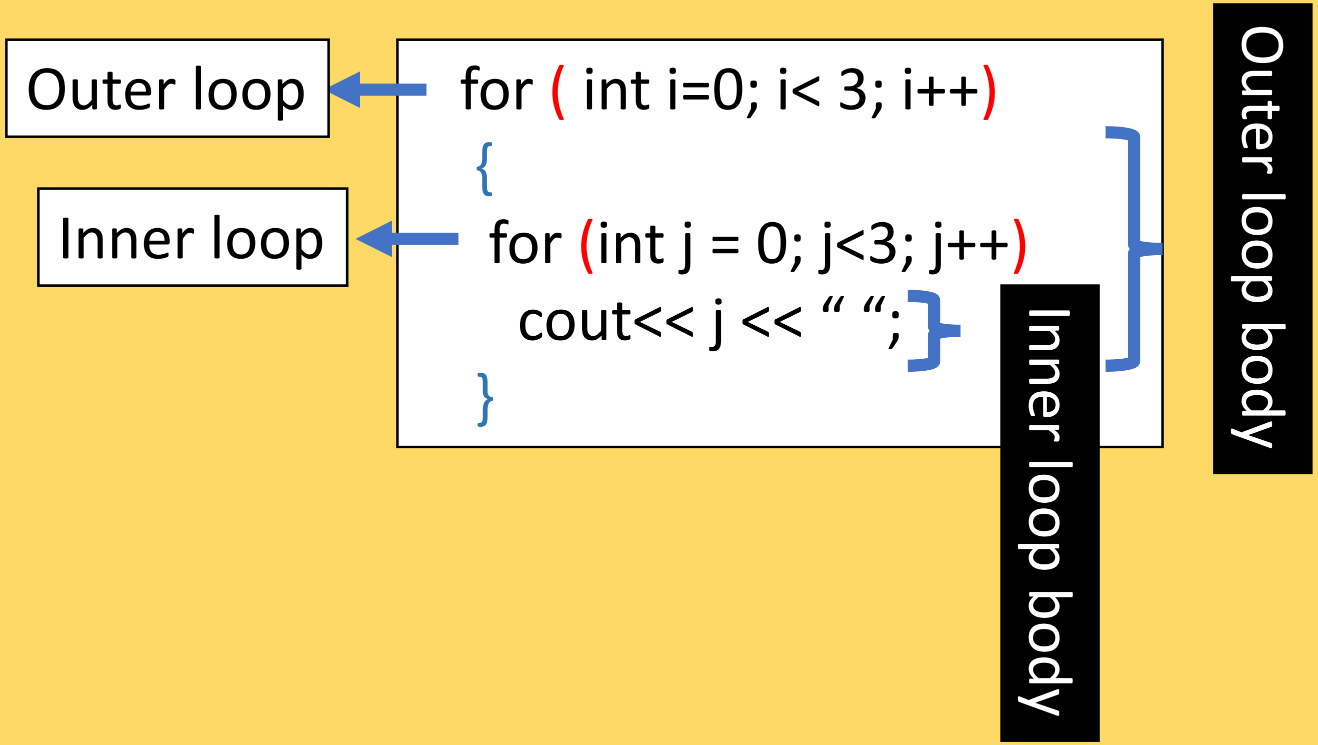
Nested Loops

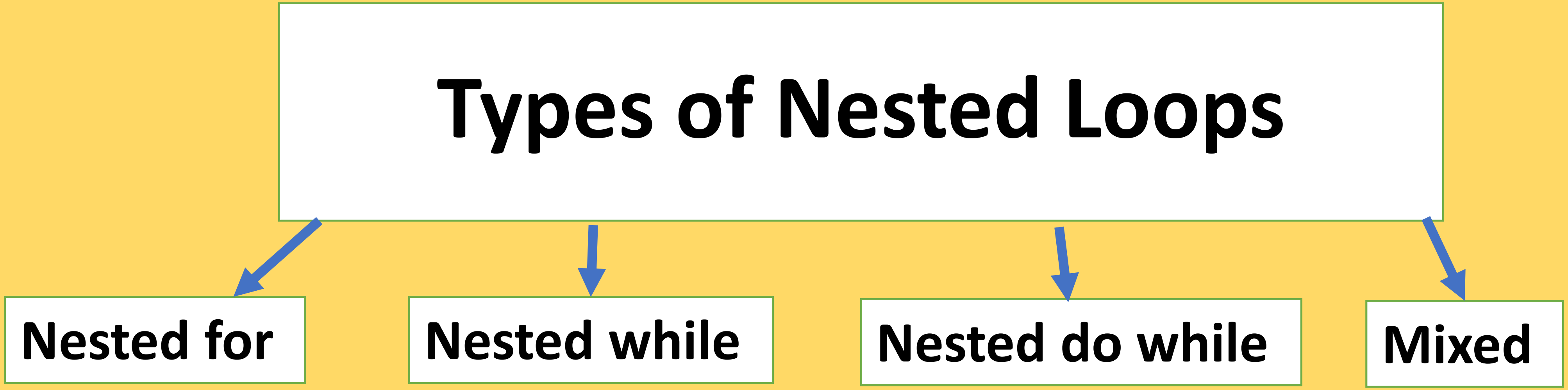
- What is a nested loop?
- Types of nested loops
- Working of a nested loop
- Example Programs

What are nested loops?

- A nested loop is a loop inside the body of another loop.
- The nested loop is known as the **inner loop** and the loop in which it is nested is known as the **outer loop**



```
for ( int i=0; i< 3; i++)
{
 for (int j = 0; j<3; j++)
   cout<< j << " ";
}
```



Types of Nested Loops

Nested for

Nested while

Nested do while

Mixed

Nested for loop

```
for ( initialization; test; update) ← outer loop
{   //outer loop statements;
 for( initialization; test; update)← inner loop
  {
      //inner loop statements;
  }
}
```

Nested while loop

```
while ( test)   ←    outer loop
{   //outer loop statements;
 while( test)  ←    inner loop
  {
     //inner loop statements;
  }
}
```

Nested do …. while loop

```
do      ←  outer loop
{   //outer loop statements;
  do    ←   inner loop
   {
      //inner loop statements;
   } while (test);

 } while(test);
```

Working of nested loops : example 1

For each iteration of the outer loop, the inner loop is completely executed.

So for example for the following nested loop:

```
for(int i=0; i<3;i++)
for(int j=0; j<3;j++)
cout<<j<<“ ” ;
```

the statement, cout<<j<<“ ” ,will execute 3X3=9 times. For each iteration of outer loop(i = 0, 1, 2) , inner loop will execute 3 times(j= 0, 1, 2).

Also note that there are no brackets. This is because both loops contain single statements; outer loop contains inner loop and inner loop contains single cout statement.

Working of nested loops : example 2

If the loop contains multiple statements then there is a need for brackets.

For the following nested loops:

```
 for(int i=0; i<2;i++)
{ cout<<“\n”;
 for(int j=0; j<3;j++)
 cout<<j<<“ ” ;
}
```

the statement, cout<<j<<“ ” ,will execute 2X3=6 times. For each iteration of outer loop(i = 0, 1) , inner loop will execute 3 times(j= 0, 1, 2).

There are brackets in the above nested loop. This is because the outer loop contains two statements; one cout<<“\n” and the inner loop . The inner loop contains single cout statement thus there are no brackets for inner loop.

Working of nested loops : example 3

If the loop contains multiple statements then there is a need for brackets.

For the following nested loops:

```
 for(int i=0; i< 4;i++)
{ cout<<“\n”;
 for(int j=1; j<= 2;j++)
 {cout<<j<<“ ” ;
 cout<<“inner loop”;
}}
```

There are brackets in both the above nested loop. This is because the outer loop contains two statements; one cout<<“\n” and the inner loop . The inner loop contains two statements 1.cout<<j<<“”and cout <<“inner loop “.

For each iteration of outer loop(i = 0, 1,2,3) , inner loop will execute 2 times(j= 1, 2).   So total no of times= 4X2=8

Working of Nested Loops

```
       1        2        8
for(int i= 0 ; i< 2;   i++)
{        3
  cout<<“\n”;

         4          5       7
  for(int j = 0;   j<2;  j ++)

          6
   cout<< j << “  ”;
}
```

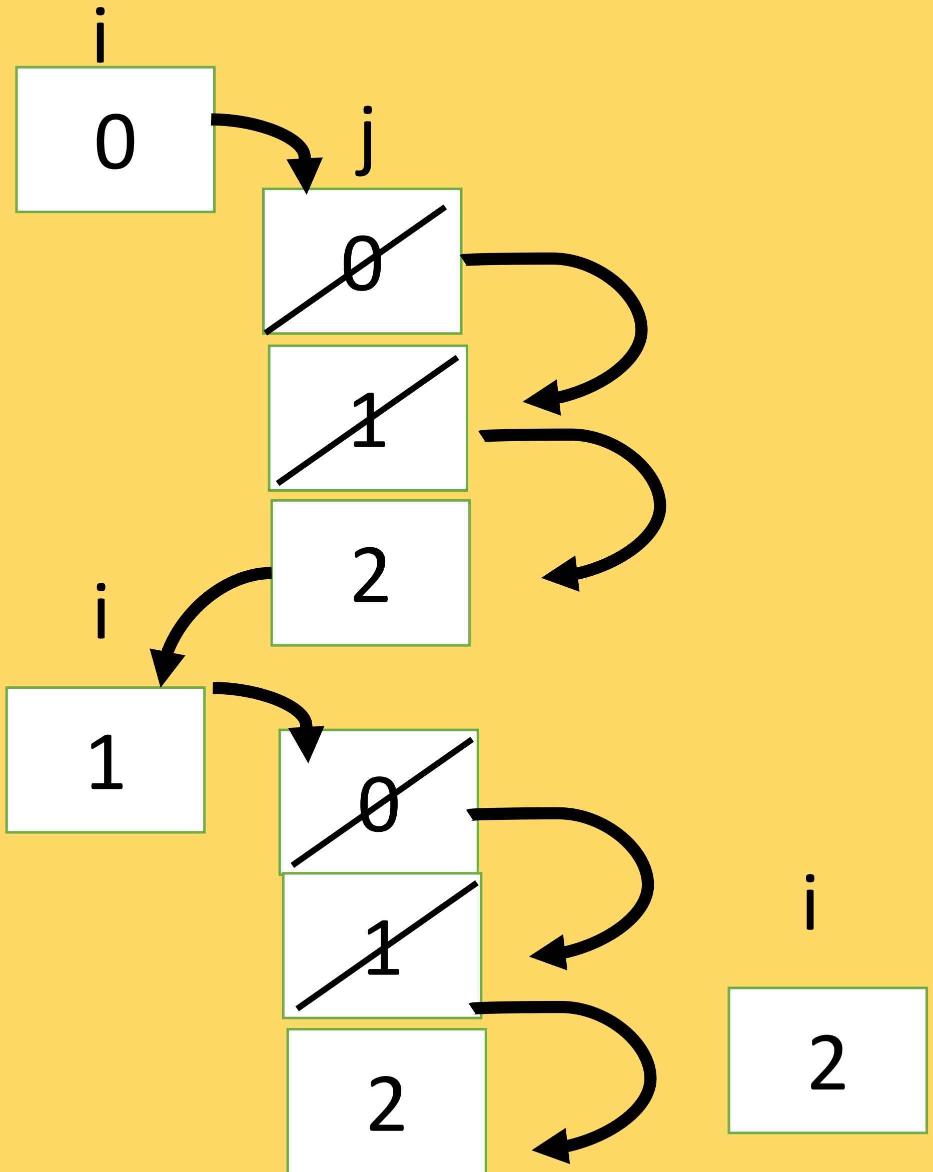
1. STEP 1:The outer loop is initialized with value of i as 0
2. STEP 2: Value of i is tested, since the condition is true(i<2), the loop is entered
3. STEP 3:A newline is displayed (cout<<“\n”;). This is part of outer loop.
4. STEP 4 :The control goes to inner loop, where j is initialized with 0
5. STEP 5: Value of j is tested, j<2 is true, inner loop is entered
6. STEP 6: The statement cout << j << “”is executed, value of j is displayed
7. STEP 7: The value of j is incremented.
8. Now STEPs 5, 6 and 7 are repeated till the condition ,j<2, becomes false.
9. When value of j is 2, control comes out of inner loop.
8. STEP 8: Now the control goes to outer loop update statement, i is incremented.
9. STEPs 2 – 7 are repeated. The steps are repeated for value of i = 1 This continues till value of i becomes 2. Then the outer loop is terminated.

Working of nested loops

```
            1          2       8
for(int i= 0 ; i< 2;   i++)
{    3
   cout<<“\n”;

           4           5    7
   for(int j = 0;   j<2;  j ++)

           6
    cout<< j << “  ”;
}
```



Output:

```
0 1
0 1
```

Example Programs

Program: To print the pattern

```
#include<iostream.h>
void main()
{
for( int x = 1; x <=3; x ++)
  {
 for ( int y = 1; y <= x; y ++)
    cout<<y <<"\t";
    cout<< "\n" ;
   }}
```

Output

```
1
1       2
1       2       3
```

Explanation

1.Outer loop : 1st Iteration x=1

1.1 Inner Loop : 1st Iteration,y=1 cout<<y<<"\t

Increment y , y=2, check y<=x=false **Inner loop over**

2. Outer loop :2nd Iteration x=2

2.1 Inner loop : 1st Iteration ,y=1 cout<< y<<"\t"

2.2 Inner loop : 2nd Iteration, y=2 cout<<y<<"\t"

Increment y , y=3, check y<=x=false **Inner loop over**

3. Outer loop :3rd Iteration x=3

3.1 Inner loop : 1st Iteration y=1, cout<<y<<"\t"

3.2 Inner loop : 2nd Iteration, y=2, cout<<y<<"\t"

3.3 Inner loop : 3rd Iteration , y = 3, cout<<y<<"\t"

Increment y , y=4, check y<=x=false **Inner loop over**

increment x, x= 3, outer loop over

Program :Program to find the divisors of numbers entered.

```
#include<iostream.h>
#include<conio.h>
void main()
{   int n, k;
   cout<< “Enter the number of integers :”;
   cin >> n;
   for( int i = 0; i < n; i ++)
  {  cout<< “Enter the number of  whose divisor are to be found”;
     cin >> k;
     cout << “\n The divisor are :”<<”\n”;
      for ( int j = 1; j <= k/2; j ++)
           if( k%j == 0)
             cout<< j <<”\t”;
}
```

Output

```
Enter the number of integers : 2
Enter the number whose divisor is to be found : 6
The divisors are :
1    2    3
Enter the number whose divisor is to be found : 15
The divisors are :
1    3    5
```

EXPLANATION:
The outer loop keeps track of the number of integer a user inputs.
The inner loop calculates and displays its divisors

Program : Display the multiplication table of a number till the user wishes

```
#include<iostream.h>
#include<conio.h>
void main()
{
  int num, l;
  char ch;
  do
 {cout<<”Enter a number whose multiplication table is to be displayed:
”;
cin>>num;
cout<<”Enter the limit of upto which table is to be displayed :”;
cin>>l;
int i = 1;
while( i <= l){
{
   cout<<num<<”X”<< i <<”=”<<num*i <<”\n”;}
  cout<<”Do you wish to continue (Y/N)”<<”\n”;
  cin>>ch;   } while(ch!=’n’||ch!=’N’);}}
```

Output

```
Enter a number whose multiplication table is to
 be displayed:  8
Enter the limit upto which table is to be
displayed : 4
8X1=8
8X2=16
8X3=24
8X4=32
Do you wish to continue(Y/N)
Y
Enter a number whose multiplication table is to
 be displayed:  6
Enter the limit upto which table is to be
displayed : 5
6X1=6
6X2=12
6X3=18
6X4=24
6X5=30
Do you wish to continue(Y/N)
N
```