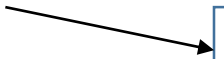
MA144: Problem Solving and Computer Programming

Lecture-8

while, do-while, for loops

Loop Control Structures

- **while**
- **do-while**
- **for**



used when an action is to be repeated for a predetermined number of times

while

Syntax

```
while(condition)
{
   statement1;
   statement2;
   statement3;
}

statement4;
statement5;
```

```
while(condition)
   statement1;

statement4;
statement5;
```

Finding GCD using while loop

```
#include<iostream>
using namespace std;

int main()
{
   int a,b, r;
   cout<<"Enter two numbers:"<<endl;
   cin>>a>>b;

   r=a%b;
   while(r!=0)
    { a=b;
      b=r;
      r=a%b;
    }
 cout<< "gcd is "<<b;
   return 0;
}
```

do-while

Syntax

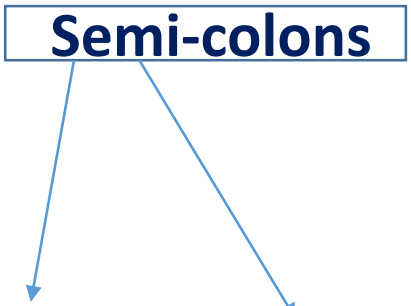
```
do
{
   statement1;
   statement2;
   statement3;

}while(condition);     //semi-colon is necessary here

statement4;
statement5;
```

Finding GCD using do-while loop

```
#include<iostream>
using namespace std;

int main()
{
  int a,b, r;
  cout<<"Enter two numbers:"<<endl;
  cin>>a>>b;

  do
   {
    r=a%b;
    if(r==0)
      break;
    a=b;
    b=r;

   }while(r!=0);

 cout<< "gcd is "<<b;
  return 0;
}
```

for

Syntax




```
for(initialization; test; increment)
{
   statement1;
   statement2;
   statement3;

}
statement4;
statement5;
```

Finding factorial of an integer using for loop

```
#include<iostream>
using namespace std;

int main()
{
   int n, i;
   double fact=1;
   cout<<"Enter an integer\n";
   cin>>n;
   for(i=1;i<=n;i=i+1)

      fact=fact*i;

cout<<fact;
}
```

Increment and Decrement Operators

Increment Operators (++) *unary operators*

- **postfix** (placed **after** the variable)

a++

- **prefix** (placed **before** the variable)

++a

Decrement Operators (--) *unary operators*

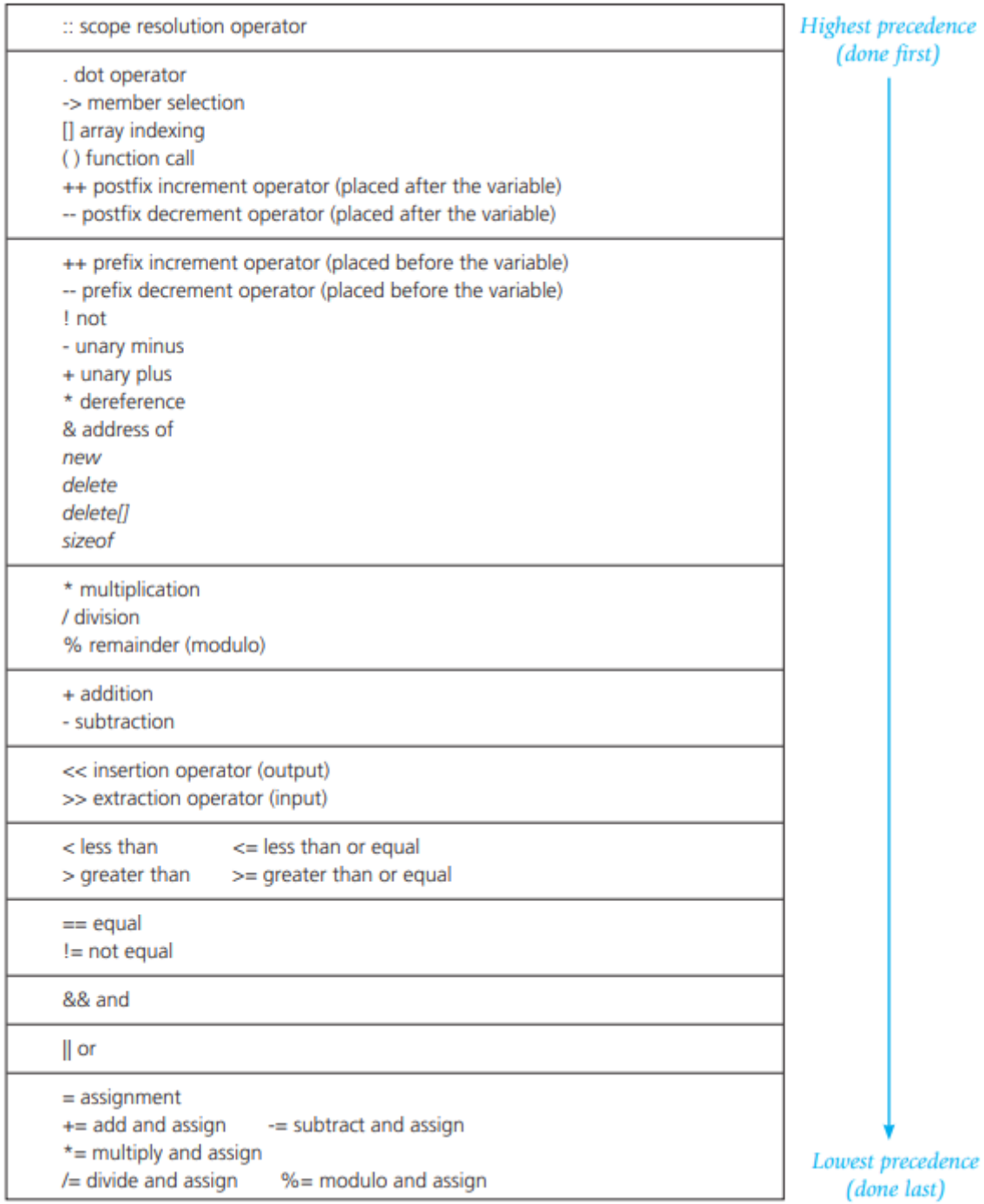
- **postfix** (placed **after** the variable)

a--

- **prefix** (placed **before** the variable)

--a

Postfix: left-right **Prefix: right-left**


:: scope resolution operator
. dot operator
-> member selection
[] array indexing
( ) function call
++ postfix increment operator (placed after the variable)
-- postfix decrement operator (placed after the variable)
++ prefix increment operator (placed before the variable)
-- prefix decrement operator (placed before the variable)
! not
- unary minus
+ unary plus
* dereference
& address of
new
delete
delete[]
sizeof
* multiplication
/ division
% remainder (modulo)
+ addition
- subtraction
<< insertion operator (output)
>> extraction operator (input)
< less than
<= less than or equal
> greater than
>= greater than or equal
== equal
!= not equal
&& and
|| or
= assignment
+= add and assign
-= subtract and assign
*= multiply and assign
/= divide and assign
%= modulo and assign
Highest precedence
(done first)
Lowest precedence
(done last)

```
#include<iostream>
using namespace std;

int main()
{   int a,i=5,b,k=5;

a=i++;
cout<<"a= "<<a<<endl<<"i= "<<i<<endl;

b=++k;
cout<<"b= "<<b<<endl<<"k= "<<k<<endl;
return 0;
}
```

a=i;

i=i+1;

k=k+1;

b=k;

```cpp
#include<iostream>
using namespace std;

int main()
{  int a,i=5,b,k=5;

a=i++;
cout<<"a= "<<a<<endl<<"i= "<<i<<endl;

b=++k;
cout<<"b= "<<b<<endl<<"k= "<<k<<endl;
return 0;
}
```

```
a= 5
i= 6
b= 6
k= 6
```

Exercise

- Write a program to find gcd using **for** loop
- Write a program to find factorial using **while** loop
- Write a program to find sum of n numbers
- Write a program to check whether the given number is prime or not
- Write a program to find all the prime numbers between 1 and the given number n
- Write a program to find lcm of two numbers
- Write a program to print a multiplication table of a given number
- Write a program to find the sum of digits of a given number