CONSTANT SPECIFIER:-

// constant specifier

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

int main( )
{
const int MAX = 2147483647;
const int N = MAX / 2;

const float KM-PER-MI = 1.60934;

const double PI = 3.14159265358979323;

return 0;

}
```

BOOLEAN VARIABLE:-

```
// boolean variable

# include <iostream>
using namespace std;

int main( )
{
bool   FLAG = FALSE

cout << "flag = " << FLAG << endl;

bool   FLAG = True

cout << "flag =" << FLAG << endl;

  return 0;

}
```

OUTPUT:-

```
flag = 0

flag = 1
```


Scanned with CamScanner

• LINEAR EQUATION SOLVER :-

ADVANCED MATH

OPERATIONS :-

⇒ Write a program to solve linear equations.

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

int main( )
{
double  a, b, c1, c2, c, d, x, y ;

cout << " Enter the co-efficient of x
and y and constant term for eq(1)= ";

cin >> a >> b >> c1 ;

cout << " Enter the co-efficient of x
and y and constant term for eq(2)= ";

cin >> c >> d >> c2 ;
```

```
double check = a*d - c*b
If (check == 0)
{ cout << "solution doesn't exist";
  return 0;
}
else
{
  x = ((c1*d) - (c2*b)) / check ;
  y. ((a* c2) - (c*c1)) / check ;
cout << "value of x = " << x << endl;
cout << "value of y = " << y << endl;
 return 0;
}
return 0;
}
```


Scanned with CamScanner

OUTPUT:-

Value of X = _____

Value of Y = _____

Enter the value of x and y and co-efficient term for eq(1) = _____

Enter the value of x and y and co-efficient term for eq(2) = _____

Relational And Equality Operators :-

Relational operators and equality operators are used to compare numerical data. Both the equality and relational operators check for a particular relationship between two numeric values and results if the relationship holds or not.

If the relationship holds, then it returns a boolean true (1)

Otherwise it returns a boolean false (0).

• EQUALITY OPERATORS :-

1. (= =) EQual :-

This operator tests if the given operands are equal: In case if they are equal it yields a boolean 'true', other wise false.

• i.e :- 2 = = 5 ;

• This will be false because 2 is not equal to 5.

2. (! =) NOT EQUAL :-

• This operator tests if the given operands are not equal. In case if they are not equal, it yields a boolean true, otherwise false.

• 2 != 5

• This will be true because

2 is not equal to 5.

RELATIONAL OPERATORS:-

1. < (Less than):-

This operator tests if the first operant is less than the second operand. If so, it yields a boolean true, otherwise false

2 < 5

This will be true because 2 is less than 5.

2. > (Greater than):-

This operator tests if the first operand is greater than the second operand. If so, it yields a boolean true otherwise false.

• 2 < 5

- This will be false because 2 is less than 5.

or

3. <= (less than equal to):-

- This operator tests if the first operand is less than or equal to second operand. If so, it yields a Boolean true, otherwise false

- 2<=5;

- This will be true because 2 is less than 5.

or

4. >= (Greater than equal to):-

This operator tests if the first operand is greater than or equal to second operator. If so, it yields a boolean true other wise false.

- 2<=5;

This will be false because 2 is less than 5.

PROGRAM:-

Q. Write a program to verify relational and equality operators.

```
# include <io stream>
using namespace std;

int main( )
{
int num1=10, num2=5, num3=-9;
int num4=5;

cout << (num1 >num2) << endl;
cout << (num2 <num3) << endl;
cout << (num2 >= num4) << endl;
cout << (num3 <= num4) << endl;
cout << (num2 == num4) << endl;
cout << (num1 != num2) << endl;

return 0;

}
```

OUTPUT:-

1

0

1

1

1

1

1

Logical Operators:-

- In case where more than one relational and equality operators simultaneously decide the final result, logical operators help in getting the simultaneous relation. Logical operators work on binary operands, their results depends on the combined states of operands.

- There are two logical operators.
  → (of our interest right now)

- AND (& &)
- OR (||)

- NOT (!) → If the value of variable is zero then its makes 1 and vice versa.

• Logical operators work on binary operands, their result depends on the combined states of operands. For the two operands X and Y (both boolean), the OR (||) operator works according to the following truth table.

X	Y	X || Y
0	0	0
0	1	0 1
1	0	1
1	1	1

→ atleast 1 true then give '1'

• AND operator (&&) works according to the following truth table

X	Y	X && Y
0	0	0
0	1	0
1	0	0
1	1	1

→ all most be true then give '1'.

• Where 0 is false and 1 is true

PROGRAM:-

Write a program by using 'OR' and "AND" command.

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

int main( )
{
int num1=10, num2=5, num3=-9, num4=6;

bool res1, res2, res3, res4;

res1 = (num1<num2)&&(num2>num3);

res2 = (num1<num2)||(num2>num3);

res3 = (num1>=num3)&&(num2>=num4);

res4 = (num1==num3)||(num2==num4);

cout<<"(num1<num2)&&(num2>num3)"<<endl;
                                  res1<<

cout<<"(num1<num2)||(num2>num3)"<<res2<<endl;
```

```
cout << "(num1>=num3)&&(num2>=num4)" << res3 << endl;
cout << "(num1==num3)||(num2==num4)" << res4 << endl;
return 0;
}
```

OUTPUT :-

(num1 < num2) && (num2 > num3) 0

(num1 < num2) || (num2 > num3) 1

(num1 >= num3) && (num2 <= num4) 1

(num1 == num3) || (num2 == num4) 0

Scanned with CamScanner

Precedence Of Arithmetic Operators:-

- **Parentheses:- ( )**

Evaluated first. If the parenthes are nested, the expression in the inner most pair is evaluated first.

- **Multiplication, Division, Modulus:-**

Evaluated second. If there are several, they are evaluated "left to right"

(*, /, %)

- first multiplication then division then modulus.
- Its type is multiplicative.

ADDITION, SUBSTRACTION:-

↳ Additive type

Evaluated last. If thee ae several, they ae evaluated "left to right"

(+ , −)

- First addition then substraction
- Its type is additive.

↳ relational operators

- < , <= , > , >=

If thee are several they ae evaluated left to right.

- Its type is relational.

↳ equality operators

- == , !=

If thee ae several they ae evaluated left to right.

Its type is equality

= → assignment operator

<< → stream insertion

>> → stream extraction

"\t" → for perfect linespace

"\n" → to end this line space.

•PROGRAM :1

// BOOLEAN VARIABLE AND LOGICAL EXPRESSION

```
# include <iostream>
using namespace std;
int main ( )
{
  bool  x = true ; // value of x is 1
int  a = 10 ;
int  b = 20 ;
 //  Equality ==
x = a==b → False // so new value of x is 0
cout << "x = " << x ;
  return 0;
}
```

OUTPUT:-

x = 0

PROGRAM : 2

```
# include <iostream>
using name space std;

int main( )

{                1          0          1          0
bool   c=true, d=false, e=true, f=true;
              1 && 0
bool   g = c && d ;   // && AND

cout << "g = " << g ;
}
```

OUTPUT:-

g = 0

// ASSIGNMENT OPERATOR AND COMPOSITE ASSIGNMENT

```
# include <iostream>
using namespace std;
int main( )
{
 int  n = 10 ;
n+ = 12 ; // n = n+12
int  m = 10 ;
m- = 5  ; // m = m-5
int  n = 15 ;
n* = 2 ; // n = n * 2
cout << "m =" << m << "n =" << n ;
   return 0;
}
```

OUTPUT:-

m = 5 n = 30

// increment and decrement

// operators ++ -- post and pre

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

int main( )
{
 int n=0;
n++ ; // n is now 1 → Post increment
++n ; // n is 2 → pre increment

int a=4, b=5 ;

int m= a+b++ ; → phlay 'a+b' hogaa
                 then post increment
cout<<"m="<<m ; (+1) nojayega

 return 0;
}
endl
```

OUTPUT:-

m= 10

Scanned with CamScanner

If-If construct

TASK : 01

Q. Write a program in which You take two number as input from user and find which number is greatest.

```
#include <iostream>
using namespace std;
int main()

{
// variable declaration
 double a,b;

//input phase
 cout<<"Enter first number :";
cin >>a;

cout<<"Enter second number :";
cin >> b;

// if-if construct
```


Scanned with CamScanner

```
if (a>b)
{
cout << "number 1 is greater";

}

if (b>a)
{
cout << "number 2 is greater";

}

return 0;

}
```


Scanned with CamScanner

TASK :02

Q. Write a program in which you take three number as input from user and find which number is greatest.

```
#include<iostream>
using namespace std;
int main()

{
// variable declaration
 int a,b,c;

// input phase
cout<<"Enter first integer:";
cin>>a;

Enter second integer:";
cout<<"
 cin>>b;

cout<<"Enter third integer:";
cin>>c;
```

```
// Decision making
// if-if construct

if (a>b && a>c)

{
cout << a << " is greater than " << b
<< " and " << c << endl;

}

if (b>a && b>c)

{
cout << b << " is greater than " << a
<< " and " << c << endl;

}

if (c>a && c>b)

{
cout << c << " is greater than "
<< a << " and " << b << endl;
}
```


Scanned with CamScanner

```
return 0;
}
```

• Assignment Q-6

```
#include <iostream>
using namespace std;
int main()
{
// variable decleration
 double a,b;
 char oper;

// input phase
 cout << "Enter two numbers: ";
 cin >>a >>b;

cout << "Enter operation ";
cin >> oper;

// Decision making
// if-if construct
```


Scanned with CamScanner

```
if (oper == '+')
 {
cout << a << "+" << b << "=" << a+b ;
 }
if (oper == '-')
{
cout << a << "-" << b << "=" << a-b ;
 }
if (oper == '*')
{
cout << a << "*" << b << "="
 << a*b ;
 }
```


Scanned with CamScanner

```
if (oper == 'e')
{
cout << a << "e" << b << "==" << a==b << endl;

}

return 0;

}
```


Scanned with CamScanner

ASSIGNMENT Q-8

```
# include <io stream>
using namespace std;
int main()

{
// variable declaration
int a,b,c;

// input phase
cout << " Input three different integers:
cin >>a >>b >>c;

// Basic mathematical operation

cout << " Sum is " << a+b+c;

cout << " \n Average is " << (float)(a+b+c)/3;

cout << " \n Product is " << a*b*c;
```


Scanned with CamScanner

```
// Decision making
// if-if construct

  if (a<b && a<c)

{
 cout << "\n Smallest is " << a << endl;

 }

 if (b<a && b<c)

{
cout << "\n Smallest is " << b << endl;

 }

if (c<a && c<a)

{
 cout << "\n Smallest is " << c << endl;

  }
```


Scanned with CamScanner

```
if (a>b && a>c)
{
cout << "Largest is " << a;
}
if (b>a && b>c)
{
cout << "largest is " << b;
}
if (c>a && c>b)
{
cout << "largest is " << c;
}
return 0;
}
```


Scanned with CamScanner

IF-ELSE CONSTRUCT

ASSIGNMENT Q.9

```
# include <io stream>
using namespace std;
int main( )

{
// variable declaration
  int a;

// input phase
 cout << "Enter any integer :";
 cin>>a;

// Decision making
// if-else construct

 if (a%2 ==0)

{
 cout << "a << "is even number";

 }
```

```
else
{
cout << "a is odd number";
}
return 0;
}
```


Scanned with CamScanner

Assignment Q.7

```
#include <iostream>
using namespace std;
int main()

{
// variable declaration
int a,b;

// input phase
 cout << "Enter two integers";
 cin >>a >>b;

// Decision making
// if-else construct

if (a>b)

 {
cout << a << " is Larger";

 }
```


Scanned with CamScanner

```
if (b>a)
{
cout << b << " is larger";
}
else
{
cout << "These numbers are equal";
}
return 0;
}
```


Scanned with CamScanner

TASK :01

Q. Write a program for the pay of a person based on hours worked.

- bonus pay start after 40 hours worked to the factor of 1.5.

```
#include <iostream>
using namespace std;
int main( )

{
 // variable declaration
 int hoursworked;
 double rate , pay;

// input phase
 cout << " Enter hours worked: ";
cin >> hoursworked;
```


Scanned with CamScanner

```
cout << "Enter rate of pay ";
cin >> rate;

// Decision making
// if -else construct

if (hoursworked <=40)

{
pay = hoursworked * rate;

cout << "pay =" << pay << endl;

 }

 else

{
pay = (hoursworked-40) * rate * 1.5 +
      hoursworked rate * 40;
cout << "pay =" << pay;

 }
 return 0;
  }
```


Scanned with CamScanner