ICPC Obour Community

Level 1 Training

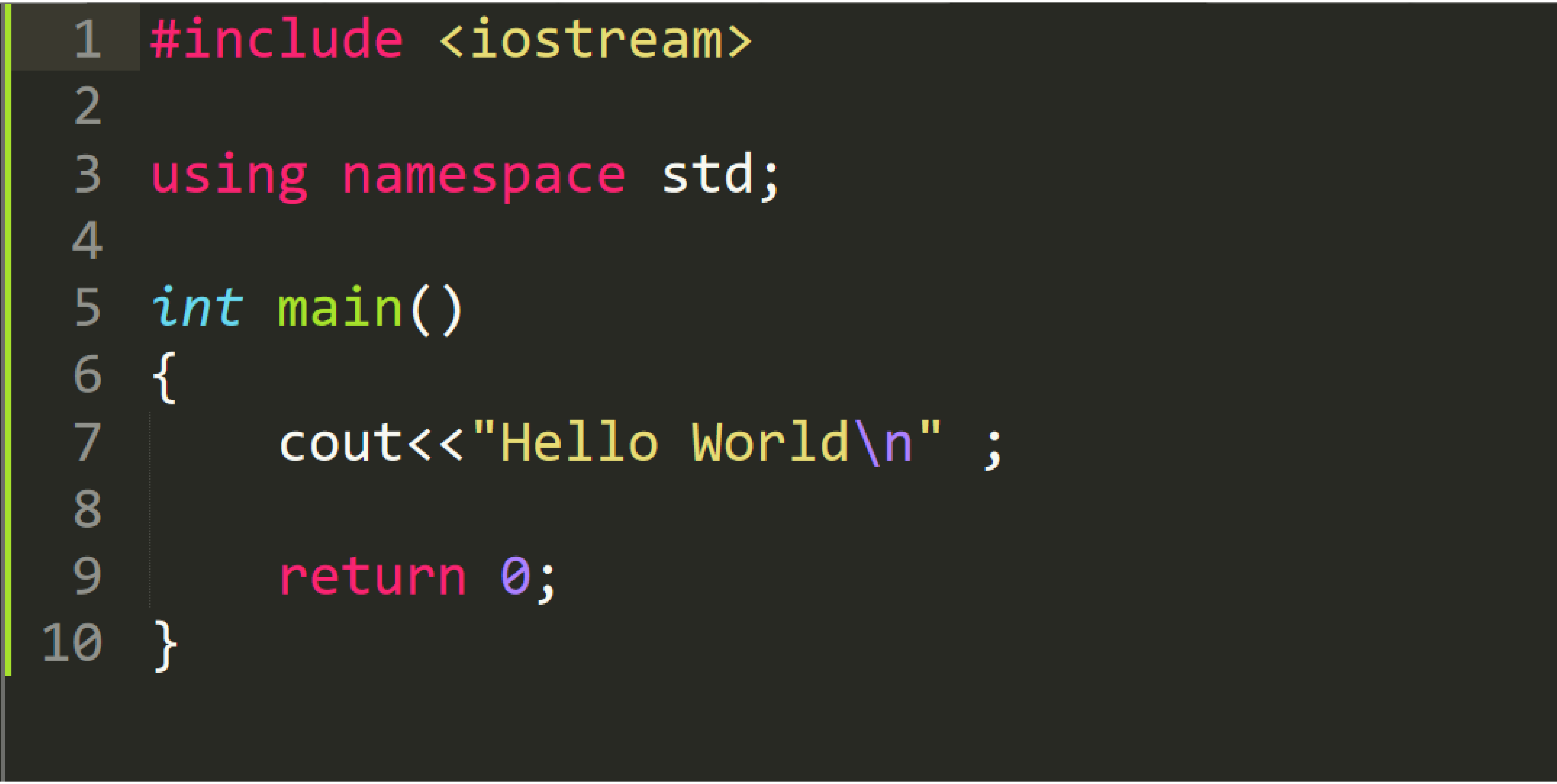
Data types

Variables

Operations

Conditions

First C++ Program



```cpp
#include <iostream>

using namespace std;

int main()
{
    cout<<"Hello World\n" ;

    return 0;
}
```

Data types

- int >> numbers → 1,3,-100 , 3487582
- float >> decimal numbers → 1.55 , -2.876
- double >> decimal numbers →1.55 , -2.876
- long long >> numbers → 5234534525643
- char >> numbers or characters → ‘a’ , 65
- string >> more than one char →“ahmed”
- bool >> true or false → true,false

Avoid → **Overflow**

Data types in C++

Data Type	Size (bytes)	Size (bits)	Value Range
unsigned char	1	8	0 to 255
signed char	1	8	-128 to 127
char	1	8	either
unsigned short	2	16	0 to 65,535
short	2	16	-32,768 to 32,767
unsigned int	4	32	0 to 4,294,967,295
int	4	32	-2,147,483,648 to 2,147,483,647
unsigned long	8	64	0 to 18,446,744,073,709,551,616
long	8	64	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	64	0 to 18,446,744,073,709,551,616
long long	8	64	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4	32	3.4E +/- 38 (7 digits)
double	8	64	1.7E +/- 308 (15 digits)
long double	8	64	1.7E +/- 308 (15 digits)
bool	1	8	false or true

Varibles

Naming rules :

1- don’t start with number like 2a

2- don’t make space like first name

3- don’t use reserved keywords like if,for

Data type   Name   Value

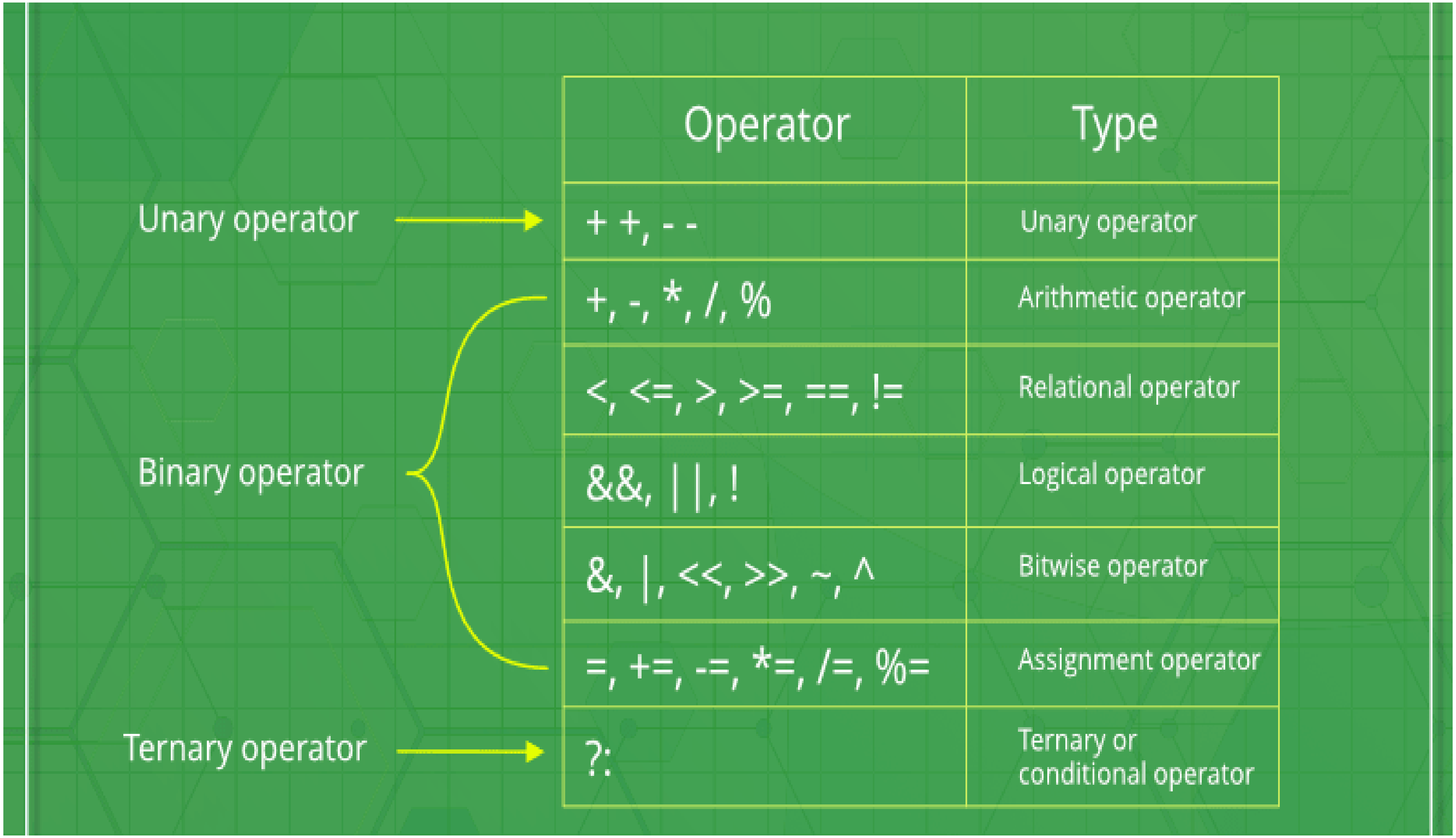
```
Int  number  = 10 ;
double number1 = 6.8 ;
char  first_c = ‘ k ’ ;
String  s = “Ali” ;
bool  ok = true ;
```

✓

```
Int  5number  = 10 ;
double first  num = 6.8 ;
char  for  = ‘ k ’ ;
```

✗

Operators in C++



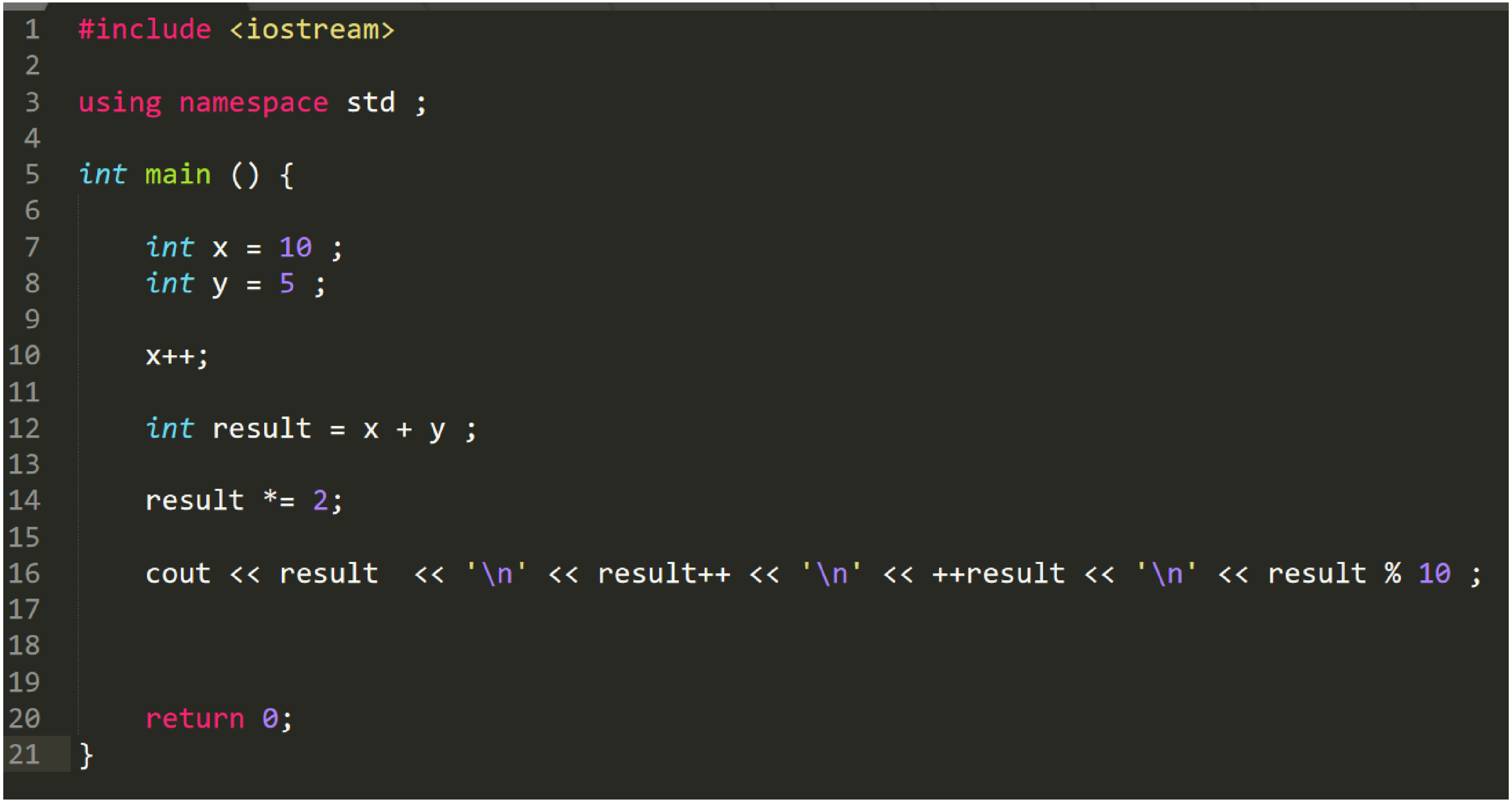
Unary operator

Binary operator

Ternary operator

Operator	Type
+ +, - -	Unary operator
+, -, *, /, %	Arithmetic operator
<, <=, >, >=, ==, !=	Relational operator
&&, | |, !	Logical operator
&, |, <<, >>, ~, ^	Bitwise operator
=, +=, -=, *=, /=, %=	Assignment operator
?:	Ternary or conditional operator

What is the output ?

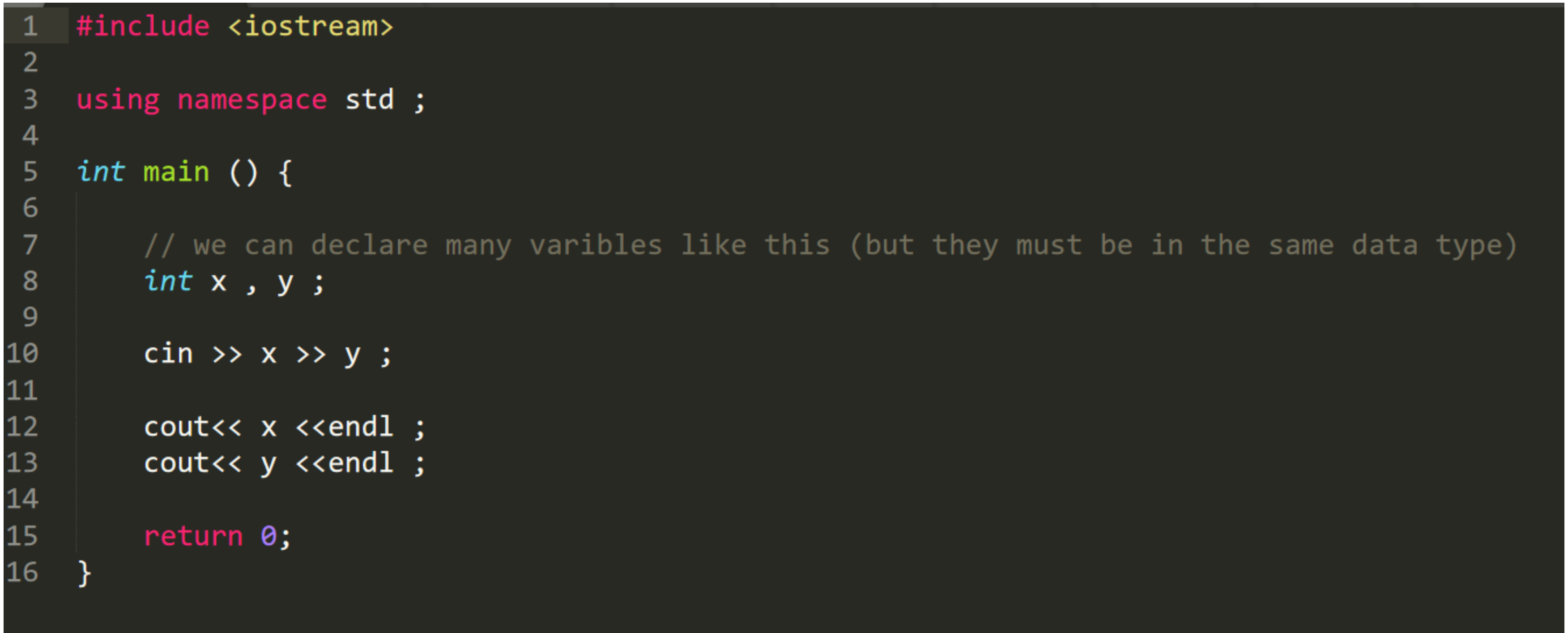


```cpp
#include <iostream>

using namespace std ;

int main () {

    int x = 10 ;
    int y = 5 ;

    x++;

    int result = x + y ;

    result *= 2;

    cout << result  << '\n' << result++ << '\n' << ++result << '\n' << result % 10 ;



    return 0;
}
```

Take input from the user

- We use cin >> to take input from the user from any data type
- To read **string with spaces** we use :

getline(cin , name) ;



```cpp
#include <iostream>

using namespace std ;

int main () {

    // we can declare many varibles like this (but they must be in the same data type)
    int x , y ;

    cin >> x >> y ;

    cout<< x <<endl ;
    cout<< y <<endl ;

    return 0;
}
```

Important notes

- To show specific number of digits after the decimal point we use :

  ```
  cout<< fixed << setprecision() << variable ;
  ```

- We can convert between data types by **casting**

- Every character has a number represents it

  we can see it in Ascii table

ASCII TABLE

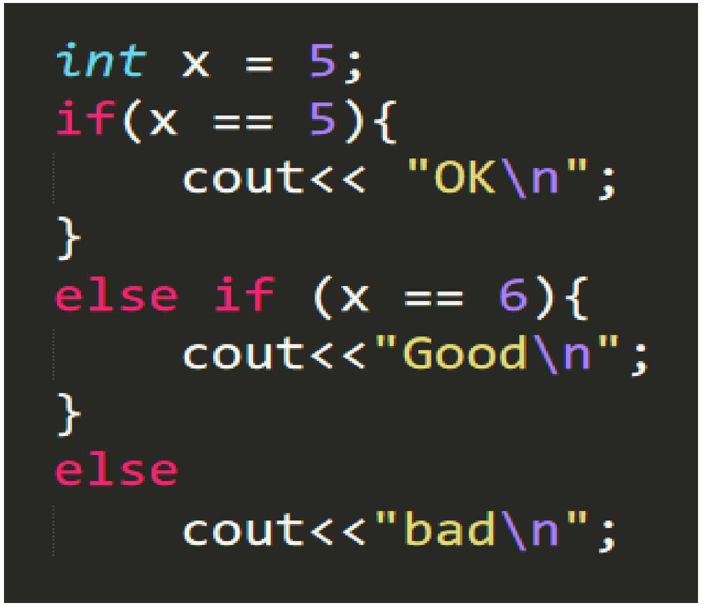
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Conditions

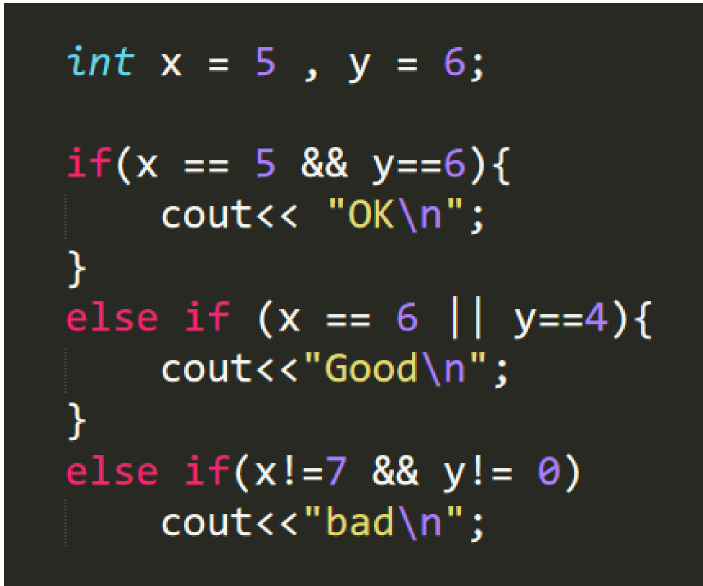
```
If ( condition ) {
   Statements….
}
```

```
If ( condition ) {
   Statements….
}
else {
}
```

```
If ( condition ) {
   Statements….
}
Else if {
}
else  if ( condition){
}
else {
}
```



```
int x = 5;
if(x == 5){
    cout<< "OK\n";
}
else if (x == 6){
    cout<<"Good\n";
}
else
    cout<<"bad\n";
```

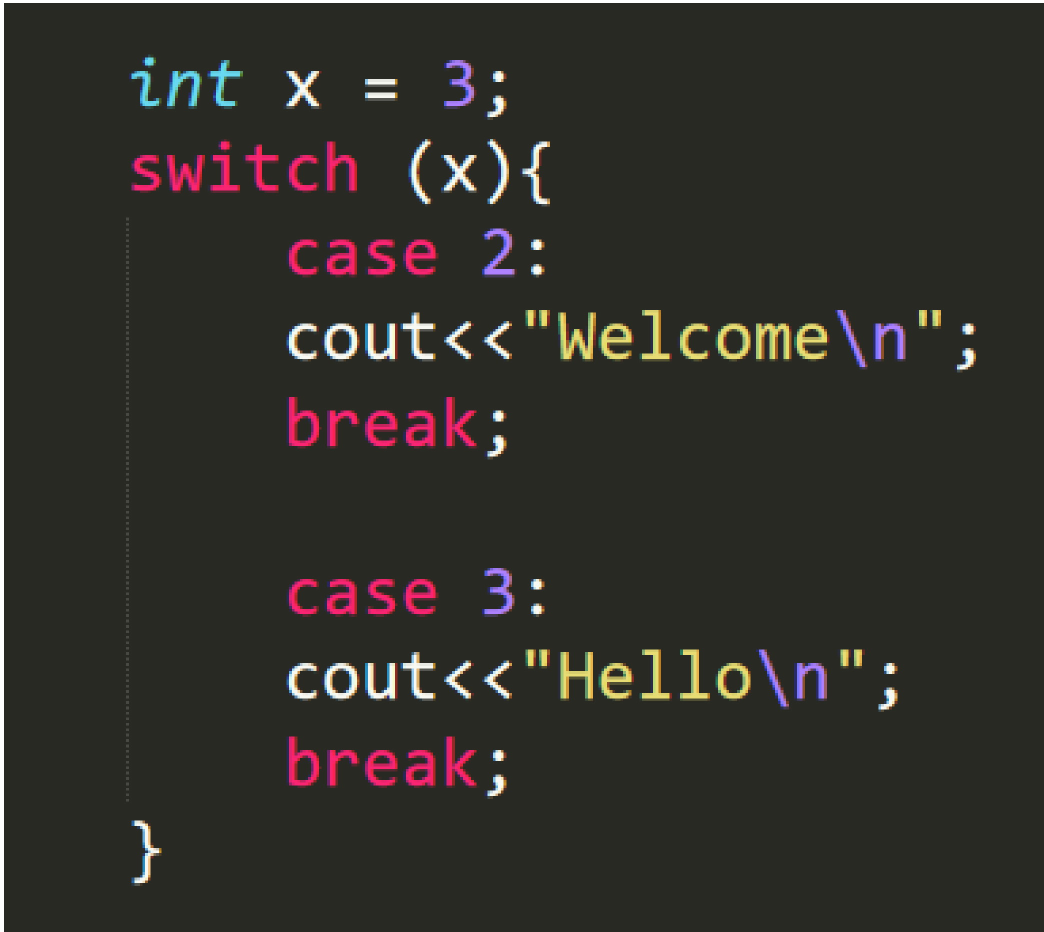


```
int x = 5 , y = 6;

if(x == 5 && y==6){
    cout<< "OK\n";
}
else if (x == 6 || y==4){
    cout<<"Good\n";
}
else if(x!=7 && y!= 0)
    cout<<"bad\n";
```

Switch case

- We can switch on a specific value like int , char
- Switch {

```
 case 3 :
     statements
     break;
 .
 .
 default :
 }
```



```
int x = 3;
switch (x){
    case 2:
    cout<<"Welcome\n";
    break;

    case 3:
    cout<<"Hello\n";
    break;
}
```