



CSC103

Computer Programming for Scientists & Engineers

Lesson (2) : Basic Elements of C++

T. Ahmed Yusuf

Binary Addition :

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

← two

$$\begin{array}{r}
 \textcircled{1} \textcircled{1} \textcircled{1} \\
 1 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 1 \quad + \\
 \hline
 1 \ 1 \ 0 \ 0 \ 0
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{cccc}
 8 & 4 & 2 & 1 \\
 1 & 1 & 0 & 1 \\
 1 & 0 & 1 & 1 \\
 \hline
 16 & 8 & 4 & 2 & 1 \\
 1 & 1 & 0 & 0 & 0
 \end{array}
 \end{array}$$

13

$$\begin{array}{r}
 11 \quad + \\
 \hline
 24
 \end{array}$$

24

(B)

$$\begin{array}{r} \textcircled{1} \textcircled{1} \textcircled{1} \\ 8 \quad 4 \quad 1 \\ 1 \ 0 \ 1 \ 1 \\ \quad 1 \ 1 \ 1 \quad + \\ \hline 1 \ 0 \ 0 \ 1 \ 0 \end{array}$$

$$\begin{array}{r} 11 \\ 7 \\ \hline 18 \end{array}$$

(C)

$$\begin{array}{r} \textcircled{1} \ 10 \ \textcircled{1} \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \quad 1 \ 1 \ 1 \ 1 \quad + \\ \hline (1 \ 0 \ 0 \ 1 \ 0 \ 1)_2 \end{array}$$

$\xrightarrow{\quad}$

$$\begin{array}{r} 11 \\ 11 \\ \hline 15 \\ (37)_{10} \end{array}$$

100

Binary Multiplication

A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} 111 \\ 101 \end{array} \begin{array}{l} \leftarrow 7 \\ \leftarrow 5 \\ \hline 35 \end{array}$$

$$\begin{array}{r} 111 \\ 0000 \\ 11100 \\ \hline 100011 \end{array} +$$

$$\begin{array}{r} 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \\ 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \end{array}$$

35

Chapter 2 : Basic Elements of C++

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

heading

```
int main ()
```

```
{
```

```
    cout << "Hello World" ;
```

```
    return 0;
```

```
}
```

beginning of the
body of the
function main

include <iostream>

Allows us to use the predefined object

cout → to generate output.

using namespace std;

allows
prefix

std::

std::cout

std::endl

<<

Stream insertion operator

Comments :

→ Comments can be used to identify the authors, give a brief explanation of the program.

→ Comments used for the reader, not for the compiler.

two comments types

Single line comments
(//)

Multiple line
Comments
(enclosed between

/* you can include
comments here
*/

Special Symbols

Mathematical
Symbols

+ - * / %

Punctuation
marks

; ? , .
← end a statement

tokens

<= != == >=

Reserved words (Keywords)

int, float, double, char, void, return

Identifiers

Identifiers are names of things that appear in programs such as variables, constants and functions.

Rules for identifiers :

→ Consists of letters, digits and the underscore character (`_`).

→ must be begin with a letter or underscore

`_num`

`Num`

is not same as

`num`

Case Sensitive

illegal identifier

first name

1number

num1 + num2

Spaces are not allowed

Can't begin with a digit

+ is not allowed

legal identifiers

first

firstName

Counter1

Which of the following is not a valid variable name declaration?

(a) `int -b3;`

(b) `int 3-b;`

(c) `int b-3;`

(d) `int -3b;`

data Type

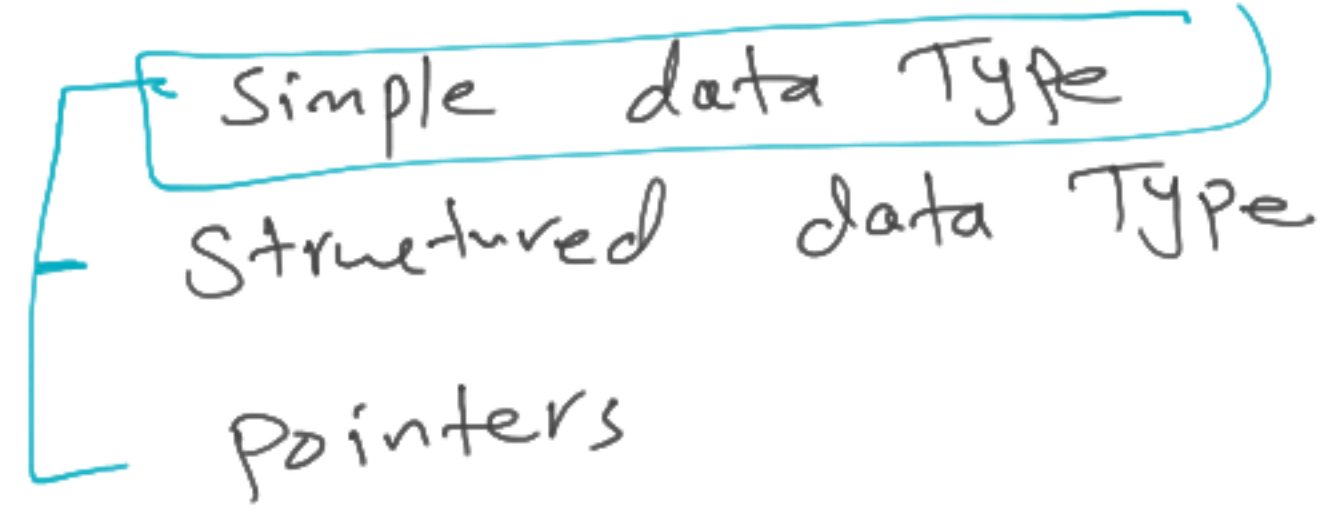
Variable Name

not valid

Which of the following is not a valid variable name:

- (a) int number;
- (b) float rate;
- (c) int variable_count;
- (d) int main();

C++ data Types



Simple data Type

int

2

80

1124

double (float)

3.14 ,

0.81

50.6

character (char)

' '

'A' ,

'*'

'a'

String

"

↖ " "

end line
cout << 2 + 8 << endl;
cout << "Multiplication:" << 2 * 3 << endl;
cout << 8 / 3 << endl;
cout << 5 / 2.5 << endl;
cout << 11 % 3 << endl;

mod
3

3
11
9

2

cout << 3 % 11 << endl;

output

10
Multiplication: 6

2

2.5

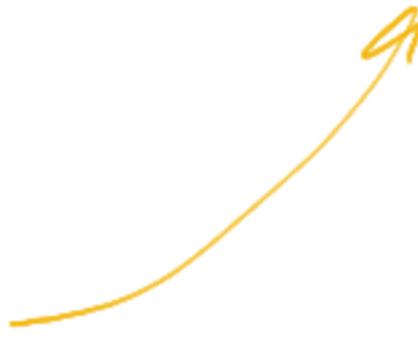
2

3

$*, /, \%$ has higher level than $+, -$

operations are performed from left to right for same level operators.

$$\text{Cout} << (3 * 7) - 6 + ((2 * 5) / 4) + 6$$
$$21 - 6 + \boxed{10 / 4} + 6 = 23$$

$$15 + \underset{2}{2} + 6 =$$


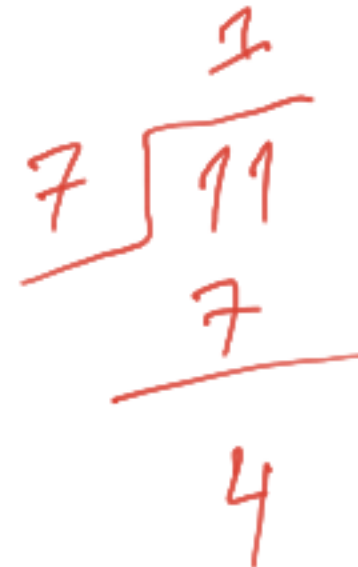
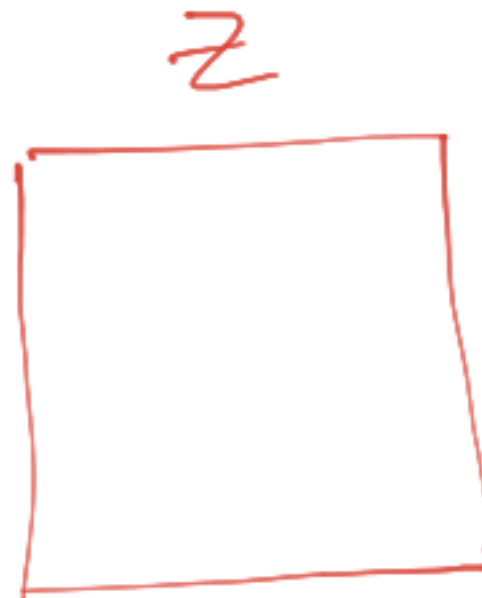
```

int x = 0;
x = 0x * 22 + 2.55 / 2.0 + 10;
cout << x << " "; space
int z = 411 % 7 + 4 / 16;
cout << z;

```

output

12 4



Increment and decrement operators :

Increment :

pre-increment

$++a ;$

post-increment

$a++ ;$

```
int x = 5 ;
```

```
cout << x++ << endl ;
```

```
cout << x << endl ;
```

output

5
6

```
int x = 5 ;
```

```
cout << ++x << endl ;
```

```
cout << x << endl ;
```

output

6
6

Decrement :

pre-decrement : $--a;$

post-decrement : $a--;$

int $n=1, m=2, p=3;$

$n++=2;$

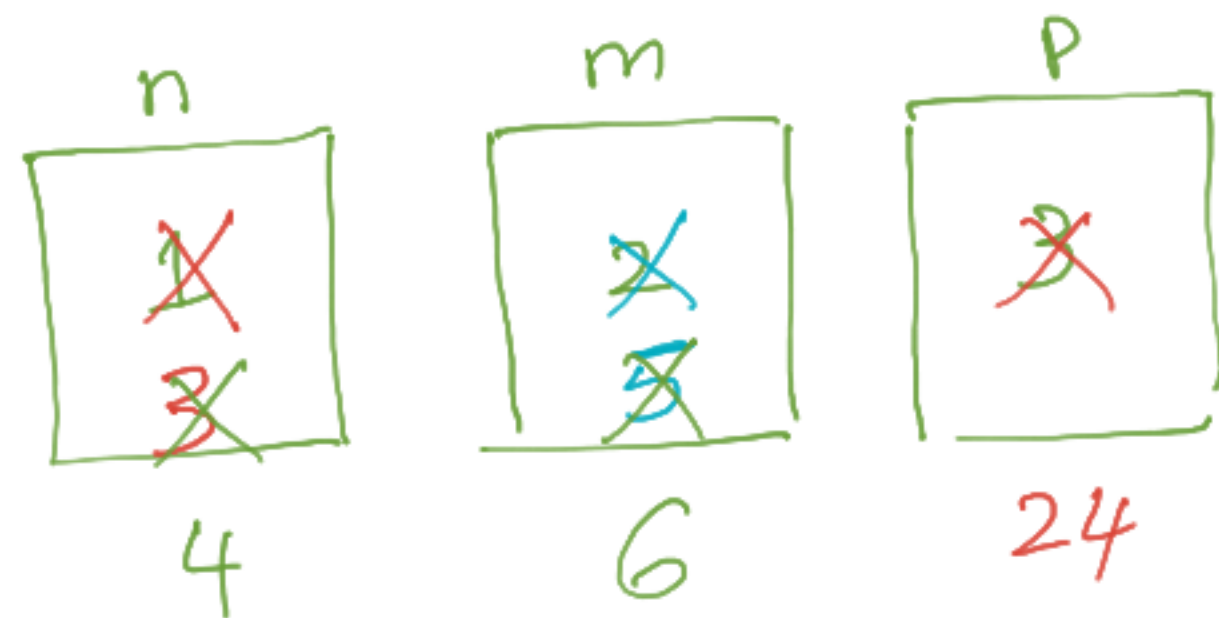
$m+=3;$

cout << n << m << p;

$p = ++n * ++m;$

$p = 4 * 6 = 24$
cout << n << m << p;

$$m = m + 3 \\ 2 + 3 = 5$$



output

3534624