

PROGRAMMING LANGUAGES 1

Variables and Operators



AREA OF CIRCLE

```
#include <iostream>
// #define PI 3.14
using namespace std;
int main()
{
    double r, area;
    const double pi = 3.14;
    cout << "Enter radius: ";
    cin >> r;
    area = pi * r * r;
    cout << "Area of circle = " << area << endl;
    return 0;
}
```

VARIABLES

- Variables in C++ must be defined in advance
- Each variable must have a **name** and a **data type**
- Available data types include int, char, float, double, ...

```
double r; // Define a variable named r of type double
int x;    // Define a variable named x of type integer
```

CONSTANTS

- Special type of variables in C++
- Defined by adding **const** before the variable definition
- Must be initialized
 - ✗ `const double pi;`
 - ✓ `const double pi = 3.14;`

VARIABLE NAMES

- Can only contain letters (a-z, A-Z), digits (0-9), and underscore (_)
- Must start with a letter or _, i.e. no digits at start
- Cannot be a C/C++ keyword, e.g. main, int, for, ...
- Should be descriptive, e.g. `radius` is better than `r` or `x`

NAMING CONVENTIONS

- Variable (and function) names should start with **small** letters or `_`, e.g. `radius`
- Multiple-word variable names should use **camelCase**, e.g. `circleArea`

DATA TYPES & MODIFIERS

- Data types

- int
- char
- float
- Double
- bool

Modifiers

- long
- short
- signed
- unsigned

DATA TYPES & MODIFIERS

Datatype	Description	Size	Range
int	(Signed) integer	8 bytes	$-2^{63} \rightarrow 2^{63} - 1$
unsigned int	Unsigned Integer (0, +)	8 bytes	$0 \rightarrow 2^{64} - 1$
short	Short integer	4 bytes	$-2^{31} \rightarrow 2^{31} - 1$
long	Long integer	8 bytes	$-2^{63} \rightarrow 2^{63} - 1$
char	(Unsigned) character	1 byte	$0 \rightarrow 255$
signed char	Signed character	1 byte	$-128 \rightarrow 127$
Float	Floating point	4 bytes	$\pm 3.4e \pm 38$ (~7 digits)
double	Double-precision floating point	8 bytes	$\pm 1.7e \pm 308$ (~15 digits)
long double	Long double-precision float	8 bytes	$\pm 1.7e \pm 308$ (~15 digits)
bool	Boolean	1 byte	true/false

sizeof()

- Built-in function to find size of variables in bytes

```
cout << "Size of int = " << sizeof(int) << endl;
```

TYPE CONVERSION & CASTING

```
int x = 5;
```

```
double y = x; // conversion
```

```
char c = (char) x; // casting
```

```
char c = char(x); // casting
```

ARITHMETIC OPERATORS

- Add +
- Subtract -
- Multiply *
- Divide /
- Modulus %
- Increment ++
- Decrement --

LOGIC OPERATORS

- And &&
- Or ||
- Not !

BITWISE OPERATORS

- Bitwise And &
- Bitwise Or |
- Bitwise Xor ^
- Complement ~
- Shift left <<
- Shift right >>

RELATIONAL OPERATORS

- Less than <
- Greater than >
- Less than or equals <=
- Greater than or equals >=
- Equals ==
- Not equal !=

ASSIGNMENT OPERATORS

- =
- +=
- -=
- *=
- /=
- %=
- &=
- |=
- <<=
- >>=

CONDITIONAL OPERATOR ?:

```
y = x > 0 ? x : 0;
```

INCREMENT & DECREMENT

- Pre-increment / decrement
 - `y = ++x; // Increment x by 1 first, then assign to y`
 - `cout << ++x; // Increment x by 1 first, then print new value`
- Post-increment / decrement
 - `y = x++; // Increment x by 1 first, then`
 - `cout << x++; // Print original value of x, then increment x by 1`

OPERATOR PRECEDENCE

Category	Operator	Associativity
Parentheses	()	Left to right
Post-increment/decrement	x++ x- -	Left to right
Unary	+x -x !x ~x	Right to left
Pre-increment/decrement	++x --x	Right to left
Multiplicative	x*y x/y x % y	Left to right
Additive	x + y x - y	Left to right
Shift	x<<y x>>y	Left to right
Relational	x<y x<=y x>y x>=y	Left to right

OPERATOR PRECEDENCE - CONTD

Category	Operator	Associativity
Equality	<code>x==y</code> <code>x!=y</code>	Left to right
Bitwise AND	<code>x & y</code>	Left to right
Bitwise XOR	<code>x ^ y</code>	Left to right
Bitwise OR	<code>x y</code>	Left to right
Logical AND	<code>x && y</code>	Left to right
Logical OR	<code>x y</code>	Left to right
Conditional	<code>? :</code>	Right to left
Assignment	<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code> <code>%=</code> <code>>>=</code> <code><<=</code> <code>&=</code> <code>^=</code> <code> =</code>	Right to left

SUMMARY

- Variables
- Constants
- Data types
- Operators
- Operator Precedence

DISCUSSION

Assuming initial value of $x = 5$ and $y = 10$.

What are the values of x , y , and z after the execution of the following statements?

1. $z = x++;$
2. $z = ++x;$
3. $z = 10 + y * x;$
4. $z = (10 + y) * x;$
5. $x \ll = 3;$
6. $z = x \gg 2;$
7. $z = y-- > ++x ? y : x;$

ASSIGNMENT 0

Extended until Wednesday 24-10-2018