

Department Of Computer Application

Dr. Rakesh Ranjan

BCA – Part III

C++ PROGRAMMING LANGUAGE

Keyword :- Keyword are the special reserve word given by the language represent special functions in the language for programming. It is the reserve identifiers can not used as variables and constant in the program.

Followings are the important keyword in C++

Auto break catch char class const
continue default delete do double else
enum extern float for friend goto if
inline int float long new
operator private protected public register
short signed sizeof static struct switch
template this throw try typedef union
unsigned virtual void volatile while

Token:

it is the smallest individual unit in a program separated by the compiler at the time of translation of

the program. Generally , tokens are the 1. Keyword
2. Identifiers 3. Constants 4. Strings 5. operators

2. Identifier :- Identifier is the symbol or symbolic name , taken by the programmer in program to define name of variable ,constant or function.

Rules of identifier naming / naming convention of variable

1. Identifier name may be any alphanumeric letter combination.

2. It cannot start with number but can follow the number after alphabets . 7ef →x

Ef76→ ok

3. Any reserve word of keyword , can not use in identifier name.

4. Operator symbol , not use in variable/identifier.

Z+b

5. White space (blank space) cannot use in variable . special character , . ; “ : are also not used in identifier name

6. _ (Underscore may use in variable name)

Abc_cde →ok Principal_amount

7 . in c++ identifier name may be 255 character but it should be small and meaningful.

Variable :

variables in program is the identifier whose meaning vary during program execution .

Constant : Constants are the program component whose meaning is always fixed , do not vary during program execution.

Eq. A, 123 , “rakesh Ranjan”

Note :- Type of variable and constant → As the data type in c++, so Corresponding the variables and constants.

Data type :

Data types are the keyword which is used to define nature and behavior or variable or constant in the

program . nature and behavior means : what will be the size of variable and constant and the range of data value to be stored .

Followings re important data types in C++:

1. Predefined data type : data types given by language is known as predefined data type –
 - a. Basic data type
 - b. Modifier / Qualifier

Basic data types are the data type which is used individually to define variable and constant with any other help

Followings are basic data type :

Integer data type : data type which define variables to store integer data → “int” → 2 byte (16-bit) → -32768 to + 32767

Floating point data type : data type define variable to store real as wel as fractional data value .

→ “float” → 4 byte (32-bit) → $\pm 3.4 \times 10^{\pm 31/38}$ → 0. 000000 (accurate result)

→ “double” → 8 byte (64-bit) → $\pm 1.7 \times 10^{\pm 308}$ → It store data in scientific form (mantissa + exponent)

$3.675 \times 10^5 \rightarrow 3.675$ (mantissa) or +5 (exponent)
→ 3.675 E+5

Character data type -> this data type define variable to store single character → ‘c’ → char → 1 byte (8-bit)
→ -128 to +127

String → Combination of characters enclosed within “ “ double quote is known as string . We have no any specific data type in C++ . We represent String by character Array . “ Rakesh Ranjan” → string → char name[20]

Modifier / qualifier -> It is supporting data type that enhance the nature and behavior of basic data type . it always use with basic data type.

1. Sign modifier -> modify sign storage

a. Unsigned → do not store sign value as a result range modify a 0 to 2 x maximum range .

unsigned int → 0 to 65535

unsigned float → 0 to

unsigned double

b. Signed-> by default data type is signed

—

signed int

int

2. Size modifier -> It modify the size of basic data type .

a. Long -> It increase the size by 2-byte .

Long int → 2 + 2 → 4 byte

Long float → 4 + 2 -> 6 byte

Long double → 8 + 2 → 10 byte

b. Short -> by default data type is in short format .

short int

int

2. User defined data type :- the data type defined by the user as per their requirement of the program by getting the existing data facility is known as user defined data type :-

- a. Struct → Structured data type
- b. enum -> Enumerated data type
- c. union → union data type
- d. class -> basic building data type of C++

3. Derived data type :- Data type created by the user as an extension of basic or existing data type is known as derived data type .

- a. Array -> Storing large volume of same data .
- b. pointer-> point to the memory location .
- c. function ->
- d. reference -> alias name

Variable declaration & Operators :-

Declaration of variables means define the variable in memory to store the data for further calculation. It is the process to define size in memory to store range of data .

Syntax -> data type variable name1, variable name 2
.....;

Eq. int a,b,c...; // a b c is an integer variable
float c,d,e ; // c,d,e is the float variable
double f,g,h; // f,g,h is the double variable
char p; // p is character variable

; - semicolon is the statement termination symbol

, → statement separator

//→ Single line comment .

Comment -> is the information in the program , written by programmer for informatics purpose which do not read by compiler during translation .

/* multiple line comment

-----*/

Type conversion :- generally same type of data work together. But during calculation some data are translated in similar type , temporarily .

Such conversion is known as type casting or type conversion .we have two type of casting →

1. Implicit casting -> compiler when translate the data in same type during calculation is known as implicit casting by the following rule

char → int → float
 ↓
 Double

2. Explicit casting → we forcibly convert the data temporarily in the calculation.

(datatype) var;

Operators :- Operators are the symbol used to create expression for calculation. It is responsible for any type of calculation in the program .

Followings are the type of operators :- expression => operand and operator .

1. Unary Operator -> Operator which is used to perform calculation with single operand .

Increment or decrement → ++/--

- → unary minus

++ → increase the value of variable with 1

--- → decrease value by 1

a. Prefix unary -> operator are placed before operand.(variable)

++a / --a in Complex expression ,
prefix unary first increase or decrease
itself then send data in calculation.

```
int a=30;
```

```
cout<< ++a; // 31
```

```
cout<< --a ; // 29
```

b. Postfix unary -> when operator are placed after the operand.

a++ / a-- ; It send data into calculation first then increase or decrease itself .

cout<< a++; // 30 → a=31

cout<< a--; // 30 → a=29

-(-a) → -30

2. Binary operator -> this operator always works in between of two operand .

a. arithmetic operator :- + , - , * , / , %(remainder)
→ left to right

b. Assignment Operator -> Assign value in variable (=). It work from right to left .

=

Lvalue Rvalue

Variable var, constant, expression

A+b=c ; //wrong c=a+b //ok

5= 6+7 ; // wrong c= 5+7 //ok

c. Relational operator -> check the relation and provide logical result true and false.

$=$ → equality

$>$ → greater than

$<$

\leq \geq \neq not equal to

d. Logical Operator :- this operator establish relationship between two or more relational expression and provide one logical result true / false

$\&\&$ → and operator

$\|$ → or operator

$!$ → not operator .

$\&\&$ operator provide true result when all the expression value be true otherwise false

.

(a > b) $\&\&$ (b >c) $\&\&$ (c>d)

T	T	T	→ True
T	F		→ False
F			→ False

`||` operator provide true value when any one result be true other wise false .

`(a >b) || (b> c) ||(c>d)`

T			$\rightarrow T$
F	T		$\rightarrow T$
F	F	T	$\rightarrow T$
F	F	F	$\rightarrow F$

`!` Operator provide reverse result of the expression

`!(a > B)`

T	$\rightarrow F$
F	$\rightarrow T$

e. Mixed assignment operator (Compound operator)

`+ =, - =, * =, / =, % =, & =, | =`

`A+=b $\rightarrow A = A+b$`

`A% =5 $\rightarrow A=A \% 5;$ if A=49; A= 49%5; 4`

`A-=5;`

`A /=3;`

A&=b;

.....

f. Bitwise operator → bit wise operator
work on the binary data level and provide
result after binary calculation.

& -> and -> when both bit be true then
true other wise false

15 & 5 → 1111 & 0101 →
0101 → 5

| → bit wise or -> if any bit be true then
true otherwise false

15 | 5 → 15

^ → exclusive or → dissimilar bit 1 and
similar bit 0.

15 ^ 5 → 1010 → 10 → 128 64 32
16 8 4 2 1

49 → 32+16+0+0+0+1 → 110001

123 → 1111011

-123 → + 01111011 → 10000101 → from
right leave 1st one as usual then
complement rest of bit in 2's complement
01100100 → 100 → 10011100 → -100

$>>$ → right shift operator → dissolve bit from right and add bit from left (negative then add 1 , if positive then add 0)

$100 >> 3 \rightarrow 01100\cancel{1}00 \rightarrow 00001100 \rightarrow 12$

$<<$ → left shift → dissolve bit from left and add bit from right

$100 << 3 \rightarrow 01100100 \rightarrow 00100000 \rightarrow 64$

2. Ternary operator (conditional statement Operator)

? : → It work with tree expression

- a. Condition
- b. 1st expression
- c. 2nd expression

If condition be true the 1st expression is executed otherwise 2nd expression is executed.

Syntax -> (Condition) ? 1st expression : 2nd expression ;

Ex-> (a>b) ? cout<< " A is greater " : cout<<" A is not greater " ;

We may do nesting of this operator also

`(a>b) > cout<<" A is greater" : (a<b) ? cout<<" A is smaller" : cout<<" both are equal " ;`

Other operator →

, → statement separation expression

; → statement termination operator

`sizeof` -> it return memory size of variable constant or datatype

```
int k = sizeof(double);
```

```
cout <<k ;
```

other C++ operator :-

Symbolic constant ->> we can create symbolic constant in C++ by using “const” keyword. Whose value is fixed at time of declaration but can not change next time in rest of program execution .

```
Const float Pi=3.14;
```

Pi is the symbolic constant whose value is always fixed in rest of program execution .