**Huazhong University of Science and Technology**

**"Computer Fundamentals and Programming (C++)"**

**experimental report**

**Experiment Name: Application of Expressions and Operators Experiment Hours: 10**

**Department: School of Life Science and Technology Major: Information Management and Information System (Medical Information)**

**Class: 1601 Name: Chen Haorui Student ID: U201612696**

**Experiment date: 2016.10.13-2016.10.21 Teacher's signature:**

**1. Purpose of the experiment**

Learn to write simple sequential C++ programs ; master the application of basic data type variables and constants ; master the application of operators and expressions ; master the use of basic control structures in structured programming ; understand the use of simple input and output ; understand the role of header files ; Familiar with the development environment of Visual C++6.0 . Learn to write standard C++ console programs with Visual C++6.0 .

**2. Experimental content:**

**1. Test the value of typical expressions prepared by yourself, and strengthen the understanding of rules such as expressions and operators. And record it on the lab report, and give a reasonable explanation for the results. (not less than 10)**

1. int a = 5 , b = 7 , c = 4 , d = 9 ; c %= c \* b % a / d ; Result: error

The result of the divisor operation on the right side of the equal sign is 0

2. const int d = 2; int a = 6 , b = 3 , c = 7 ; ( ); cout << a << "\n" ;

① a = --b \* d \* ++c ; result: 32 ② a = --b \* d \* c++ ; result: 28

③ a = b-- \* d \* ++c ; result: 48 ④ a = b-- \* d \* c++ ; result: 42

If the value after addition/subtraction is used in the self-increment and self-subtraction operation, the self-increment and self-subtraction sign must be preceded, otherwise it is invalid

3.

int c = 2 , d ; char a = 'a' , b = '1' ; c += a >= b ; d = c == b; cout << c << d << "\n" ;

Result: c = 3 , d = 0 Relational operations take precedence over assignment operations

4.

int a , x = 0 , y = 2 , z = 3 ; a = x || y >= z && ( z = z + z ) ; cout << a << endl ;

Result: a = 0 greatly simplifies calculations through evaluation optimization

5. int a = 3 , b = 0 , c = 5 , z ; ( ) ; cout << z << " " << c << endl ;

① z = a > b && a < c++ ; result: z=1, c=6 ② z = a < b && a < c++ ; result: z=0, c=5

③z = a > b||a < c++ ; Result: z=1, c=5 ④ z = a < b||a < c++ ; Result: z=1, c=6

Calculations can be greatly simplified by evaluation optimization

6.

double x; int a = 3 , b = 0 , c = 5 , z ; c += ++b + sizeof (x) ;

z = --b || ++a || c; cout << z << " " << c << " " << a << endl ; result: z = 1 , c = 14 , a = 4

7.

char c1='a ',c 2='b'; c1=c1-'a'+'A'; c2=c2-'a' + 'A'; cout <<c1<<' '<<c2<< endl ;

: c1= ' A ' ; c2= ' B'The characters participate in the operation in the form of ASCII in the machine

8.

int neusea [ 6]; neusea [1] = 1; neusea [2] = 2; neusea [3] = neusea [2] + neusea [1] ;

neusea [ 4] = neusea [3] + neusea [2]; neusea [5] = neusea [4] + neusea [3] ; neusea [6] = neusea [5] + neusea [4] ;

cout << neusea [6] << "\n" ;

Result: 13 (memory value cannot be written?) Fibonacci sequence

9.

int a = 6 , b = 2 , c = 9 ; int \*s1 , \*s2 , \*s3 ; s1 = &a; s2 = &b; s3 = &c; cout << a << " " << b << " " << c << endl ;

cout << \*s1 << " " << \*s2 << " " << \*s3 << endl ;

Result: a = s1 = 6; b = s2 = 2; c = s3 = 9

10.

int a = 5 , &b = a ; a = a \* a; cout << a << " " << b << endl ; b= b / a ; cout << a << " " << b << endl ;

a and b as the same variable by reference

**3. Algorithm description (use words or flowcharts to explain the algorithm of the above programming questions )**

1. Direct output, the output format is controlled by ' \t ' , ' \n ' and endl .

2. The character variable can be converted by ( int ) symbol , and the ASCII code value of the character can be displayed.

3. After adding the header file <cmath> or < math.h > , you can insert function operations. Note that the computer can use operators to perform operations. At present, I have only learned relational symbols, assignment symbols, and four arithmetic operations.

4. After adding the header file <iomanip> , you can control the output format, including data format control (usually iosetflags ) and typesetting control ( setw )

5. Computer operations are performed according to the priority of operators, and operators of the same level are executed at the same time. Pay attention to the requirements of operators like % on data types.

**4. The main problems and solutions encountered in this experiment (record truthfully and in detail)**

Compile Error:

1. Forget to add a semicolon; 2. a = b - - c ++ cannot be executed; 3. a > = b ;

Logic error: the variable forgets to assign a value and directly participates in the operation

Solution:

1. Add a semicolon, practice more, and pay attention to details.

2. An operator is added between b - - and c++, because after the execution of the equation, it actually becomes a = bc , and the computer cannot recognize it. 3. Relational operators and all operators composed of two symbols cannot have spaces between them , otherwise the computer cannot recognize

**5. Suggestions for improving the content, methods and means of this experiment, as well as experimental experience**

1) Which knowledge points have been mastered:

the division operation are both integers, the result is rounded according to the truncated method; the basic writing of arrays, pointers, and references.

2) Which knowledge points are difficult:

The priority of operation execution is sometimes not well remembered, and the functions of each control symbol will be confused when controlling the output format.

3) Suggestions for lectures: slow down 4) Remedial suggestions for lack of knowledge: practice more