ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Week 5: Expressions

Topic of this week

- Expression
  - mathematic operator
  - boolean operator
  - conditional expression
- Programming Exercise

Expression and Operations

- Arithmetic Operators
  - Addition +
  - Subtraction -
  - Multiplication *
  - Division /
  - Modulation %
- Example
  - `fag = x % y;`
  - `c = a – (a/b)*b;`
  - `sum = var1 + var2 + var3;`

Expression and Operations

- Operator precedence
  - Some arithmetic operators act before others (i.e., multiplication before addition)
  - Use parenthesis when needed
- Example: Find the average of three variables: **`a`**, **`b`** and **`c`**
  - Do not use: `a + b + c / 3`
  - Use: `(a + b + c ) / 3`

Expression and Operations

- Rules of operator precedence:

Operator(s)	Operation(s)	Order of evaluation (precedence)
`()`	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they are evaluated left to right.
`*`, `/`, or `%`	Multiplication Division Modulus	Evaluated second. If there are several, they re evaluated left to right.
`+` or `-`	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Decision Making

- Executable statements
  - Perform actions (calculations, input/output of data)
  - Perform decisions
    - print "pass" or "fail" for a given value of grade
- `if` control structure
  - Simple version in this section
  - If a condition is true, then the body of the `if` statement executed
    - `0` is false, non-zero is true
- Keywords
  - Special words reserved for C
  - Cannot be used as identifiers or variable names

Decision Making

- Relational Operators
  - Less than `<` a < 5
  - Less than or equal `<=` a <= b
  - More than `>` a > b + c
  - More than or equal `>=` a >= b + 5
  - Equal `==` a == -6
  - Not equal `!=` a != 0

Decision Making

Keywords			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Example 1

```
2   #include <stdio.h>
3
4   int main()
5   {
6      int num1, num2;
7
8      printf( "Enter two integers, and I will tell you\n" );
9      printf( "the relationships they satisfy: " );
10     scanf( "%d%d", &num1, &num2  );   /* read two integers
*/ 11
12     if ( num1 == num2 )
13        printf( "%d is equal to %d\n", num1, num2 );
14
15     if ( num1 != num2 )
16        printf( "%d is not equal to %d\n", num1, num2 );
17
18     if ( num1 < num2 )
19        printf( "%d is less than %d\n", num1, num2 );
20
21     if ( num1 > num2 )
22        printf( "%d is greater than %d\n", num1, num2 );
23
24     if ( num1 <= num2 )
25        printf( "%d is less than or equal to %d\n", num1,
num2);
```

```
26
27      if ( num1 >= num2 )
28         printf( "%d is greater than or equal to
%d\n",
29                 num1, num2 );
30
31      return 0;    /* indicate program ended successfully
*/
32   }
```

```
Enter two integers, and I will tell you
the relationships they satisfy: 3 7
3 is not equal to 7
3 is less than 7
3 is less than or equal to 7
```

```
Enter two integers, and I will tell you
the relationships they satisfy: 22 12
22 is not equal to 12
22 is greater than 12
22 is greater than or equal to 12
```

Expression and Operations

- Logical Operators
  - AND **&&** (a > 0) && (b > 0)
  - OR || (a <= 0) || (b <= 0)
  - Negation **!** !(a && c)

Expression and Operations

- Bitwise Operators
  - Bitwise AND &
  - Bitwise OR (Inclusive OR) |
  - Bitwise XOR (Exclusive OR) ^
  - Left shift <<
  - Right shift >>
  - One's complement ~
- Example
  - x = 01001011 y = 00101100
  - ~x = 10110100
  - x & y = 00001000 x | y = 01101111
  - x ^ y = 01100111 x << 2 = 00101100

Expression and Operations

- Assignment Operators and Expressions
  - **`operator:`** + - * / % << >> & ^ |
  - If **`expr1`** and **`expr2`** are expressions, then

```
expr1 op= expr2
```

  - Equivalent to

```
expr1 = (expr1) op (expr2)
```

- Example
  - `x += 1;`
  - `x = x + 1;`

} Equivalent

Expression and Operations

- Conditional Expressions

```
expr1 ? expr2 : expr3
```

  - If **`expr1`** is *true* do **`expr2`**
  - If **`expr1`** is *false* do **`expr3`**

- Example
  -
```
a = 5;
b = 10;
min = a < b ? a : b;
```

Exercise 5.1

Write a program asking for 3 integers then:

- display the maximum number
- if the average value of these 3 number > 10, display the sum of 2 first numbers. In other case display the difference of the 2 last numbers.

Note: do not use if else structure in all exercises at this week

Hint

- max = a>b? a:b;
- max = max >c ? max:c;

- (float)(a+b+c)/3>10? printf(“sum of etc.. %4d”,a+b) : printf(“substraction of etc.. %4d”,b-c)

Expression and Operations

- Increment and Decrement Operators
  - Pre-increment operation `++variable`
  - Post-increment operation `variable++`
  - Pre-decrement operation. `--variable`
  - Post-decrement operation. `variable--`
- Example
  - ```
    x = 4;
    y = x++ + 5;        // x = 5, y = 9
    ```
  - ```
    x = 4;
    y = ++x + 5;        // x = 5, y = 10
    ```

Expression and Operations

- Type Cast Operator (Casting)

```
(type-specifier)  expression;
```

- Example
  - `float var1 = 2.7;`
    `int var2 = (int) var1;  //var2 = 2`
  - `(char) x;`
  - `(int) d1 + d2;`

Exercise 5.2

- Write a program that converts distances from kilometers to miles.
- Ask user to input the kilometers value then output to screen the miles value.
- 1 mile ~= 1.609344 km

Solution

```
#include <stdio.h>
#define KM2MILE 1.609344
int main()
{
        double    miles, kms;
        /* Get the distance in kilometers. */
        printf("Enter the distance in kilometers: ");
        scanf("%lf", & kms);
        /* Convert the distance to miles. */
         miles = kms/ KM2MILE;
        /* Display the distance in miles. */
        printf("That equals %.3lf miles.\n", miles);
        return 0;
 }
```

Exercise 5.3

- Run the exercise5_3.c program below to illustrate the operation of Logical operators and relational operators.
- Replace **b - a == b – c** by **a = b-c** and then explain the result.

exercise5_3.c

```
#include <stdio.h>
int main()
{
        int a = 5, b = 6, c = 7;
        puts("int a = 5, b = 6, c = 7;\n");
        printf("The value of  a > b  is  \t%i\n\n", a > b);
        printf("The value of  b < c  is   \t%i\n\n", b < c);
        printf("The value of  a + b >= c      is \t%i\n\n", a
        + b >= c);
        printf("The value of  a - b <= b-c    is\t%i\n\n", a
        - b <=b- c);
        printf("The value of  b - a == b - c  is\t%i\n\n",b-
        a==b- c);
        printf("The value of  a * b != c * c  is\t%i\n\n",a *
        b<c * c);
        returrn 0;
}
```

Exercise 5.4

- Type and compile the exercise5_4.c below, the program illustrates the operation of conditional expressions.

exercise5_4.c

```
#include <stdio.h>
int main()
{
	int n, m, abs, max;
	printf("Enter a positive or negative integer: ");
	scanf("%i", &n);
	printf("\nYou entered %i.\n", n);
	abs = n < 0 ? -n : n;
	printf("Its absolute value is %i.\n", abs);

	printf("\nEnter two integers (e.g. 1 2): ");
	scanf("%i %i", &n, &m);

	printf("\nYou entered %i and %i.\n", n, m);
	max = n > m ? n : m;
	printf("%i is the larger value.\n", max);
	returrn 0;
}
```

Exercise 5.5

- This example illustrates the **integer overflow** that occurs when an arithmetic operation attempts to create a numeric **value** that is larger than can be represented.
- Type and compile the program to see the result.

exercise5_5.c

```
#include <stdio.h>
#include <limits.h>

int main()
{
        unsigned int x = UINT_MAX - 1;
        signed int y = INT_MAX - 1;

        printf("x is an unsigned int, occupying %i
        bytes.\n\n", sizeof(x));

        printf("The initial value of x is %u\n", x);
        x++;
        printf("Add 1; the new value of x is %u\n", x);
        x++;
```

exercise5_5.c

```
    printf("Add 1; the new value of x is %u\n", x);
    x++;
    printf("Add 1; the new value of x is %u\n", x);
    printf("\ny is a signed int, occupying %i bytes.\n\n",
    sizeof(y));
    printf("The initial value of y is %i\n", y);
    y++;
    printf("Add 1; the new value of y is %i\n", y);
    y++;
    printf("Add 1; the new value of y is %i\n", y);
    y++;
    printf("Add 1; the new value of y is %i\n", y);

    return 0;
}
```

Exercise 5.6

- Write a program that requires user to input two double values stored in two variables x,y.
- Use **if** control structure to examine all the relation between x and y.

Solution

```
#include <stdio.h>
int main()
{
      double num1, num2;

       printf( "Enter two doubles, and I will tell you the
      relationships they satisfy: " );
       scanf( "%f%f", &num1, &num2  ); /* read two integers */

       if ( num1 == num2 )
           printf(  "%f is equal to %f\n", num1, num2 );

       if ( num1 != num2 )
          printf(  " %f is not equal to %f\n ", num1, num2 );
```

Solution

```
    if ( num1 < num2 )
        printf( "%f is less than %f\n", num1, num2 );

    if ( num1 > num2 )
       printf( "%f is greater than %f\n", num1, num2 );

    if ( num1 <= num2 )
       printf( "%f is less than or equal to %f\n",
               num1, num2 );

    if ( num1 >= num2 )
       printf( "%f is greater than or equal to %f\n",
               num1, num2 );

    return 0;   /* indicate program ended successfully */
}
```

Homework 1

- You are chatting with 2 boys and have to make decision what boy you should make a dating.
- Ask the boys about
  - Age:
    - (<=18): -2 points
    - 18 < age < 24: 5 points
    - >= 24: 2 points
  - Height:
    - >= your height: + 3 points
    - < your height: - 2 points
  - The boy with higher point is choosen. In the case 2 boys get the same points – make dating with them in Sartuday and Sunday.

Interface

- Login – enter your height (cm): 170
- Hi Minh:
  - How old are you? 22 (5)
  - What's your height? 160 (3)
- Hi Manh:
  - How old are you? 25 (2)
  - What's your height? 172 (5)
- Your decision:
  - Manh, are you free on Saturday?

Homework 2

- FPTShop gives the promotion on the Apple product this month.
  - If you buy >= 2 iPhone (1000 USD/iPhone): 5% discount
  - If you buy >= 2 Macbook (1500 USD/Macbook): 10% discount
- Write a program that get from the users the numbers of iPhone and Macbook, then print in detail the bill they have to pay.

Interface

- Ban muon mua bao nhieu iPhone? 4
- Ban muon mua bao nhieu Macbook ? 1
- FPT SHOP - HOA DON THANH TOAN
  - iPhone 4 x 1000 = 4000
  - Macbook 1 x 1500 = 1500
  - Net 5500
  - Discount

    iPhone 4000 * 5%= 200
  - You pay: 5500 – 200 = 5300