ME 172

Introduction to C Programming Language

Lecture 2: Operators and Conditional Statements

Md. Tusher Ahmed

Assistant Professor, Dept. of ME,BUET

Courtesy: Dr. Noor Al-Quddus

Dr. Monjur Morshed

M Jamil Hossain

Cyrus Ashok Arupratan Atis

M Aminul Hoque

Partha Kumar Das

Please go to this link, fill up the form and submit!!

This will allow us stay in touch with you and communicate!

https://goo.gl/forms/jD2ytKRGfUkDOgIx1

Operators...

An operator is a symbol that tells the computer to perform certain mathematical or logical manipulation

Arithmetic operator:

There are five(**5**) arithmetic operators in C

Operator	Name	Example
+	Addition	*a + b*
-	Subtraction	*a – b*
*	Multiplication	*a * b*
/	Division	*a / b*
%	Remainder	*a % b*

Arithmetic operators (Contd...)

- The data items that operators act upon are called ***operands***
- The operands can be integer quantities, floating-point quantities or characters

The remainder operator (%) requires that both operands be integers and the second operand be nonzero. Similarly, the division operator (/) requires that the second operand be nonzero.

Division of one integer quantity by another ***always*** results in a truncated value (i.e., the decimal portion of the value will be dropped).

If a division operation is carried out with two floating-point numbers, or with one floating-point number and one integer, the result will be a floating-point

Arithmetic operators (Contd...)

Example:

If a and b are integers

a=10 ; b=3

Expression	Value
a + b	13
a - b	7
a * b	30
a /b	3
a % b	1

Performance Test 1

- Write a C program that will divide 29765 apples to 51 buyers. Display how many apples each buyer will get and how many apples will be left (Use of arithmetic operator is a must, do not do the calculations and then print the desired output).

Time: 3 minutes!!

ANSWER

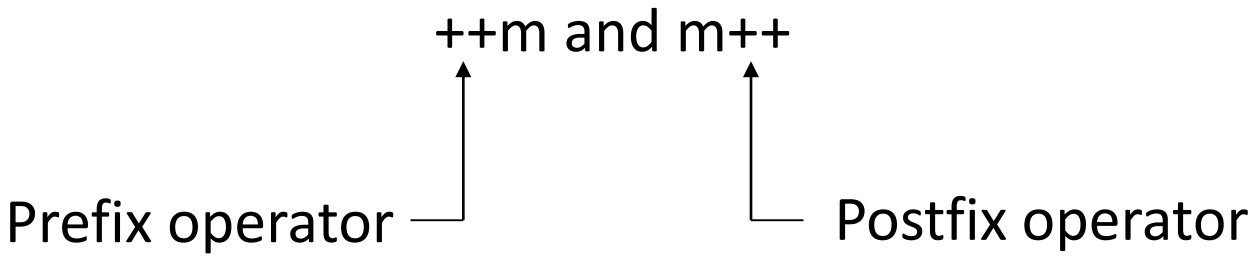
```
#include <stdio.h>

void main()
{
   int a,b,c,d;
   a=29765;
   b=51;
   c=29765/51;
   d=29765%51;
   printf("%d and %d",c,d);
}
```

```
583 and 32
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```

Increment/decrement operator

- ++a/a++ is equivalent to a=a+1
- --a/a-- is equivalent to a=a-1



The difference for built-in types is:

- ++a first increments the value of a and then returns a value referring to a, so if the value of a is used then it will be the incremented value.
- a++ first returns a value whose value is a, that is, the old value, and then increments a at an unspecified time before the next full-expression (i.e., "before the semicolon").

Increment/decrement operators(Contd...)

- x=x*a++ is equivalent to x=x*a ; a=a+1
- x=x*++a is equivalent to a=a+1 ; x=x*a

- y=y*b-- is equivalent to y=y*b ; b=b-1
- y=y*--b is equivalent to b=b-1 ; y=y*b

Increment/decrement Operators(Contd...)

- Write the following program:

```
#include<stdio.h>

void main()
{
int a=10,b=20,x;
x=a*++b;
printf("\n The value of x is: %d”,x);
}
```

For x=a*++b output:
210
& for x=a*b++ output:
200

Replace the line x=a*++b with x=a*b++

Assignment Operator

Operator	Description	Example
+=	Add AND assignment operator	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator	C /= A is equivalent to C = C / A

Relational Operators (Contd..)

Also called Comparison operators

It performs tests on their operands. They return he Boolean value . Such as:

- 1 if the statement is successful (true)
- 0 otherwise

Example	Name	Result
a == b	Equal	**TRUE** if a is equal to b.
a != b	Not Equal	**TRUE** if a is not equal to b.
a<b	less than	**TRUE** if a is strictly less than b.
a>b	greater than	**TRUE** if a is strictly greater than b.
a<=b	less than or equal to	**TRUE** if a is less than or equal to b.
a>=b	greater than or equal to	**TRUE** if a is greater than or equal to b.

LOGICAL Operators

Example	Name	Result
! a	Not	**TRUE** if a is not **TRUE**.
a && b	And	**TRUE** if both a and b are **TRUE**.
a || b	Or	**TRUE** if either a or b is **TRUE**.

a	b	a && b	a || b
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

The ? : operator

General form:

Conditional expression? Expression1 : Expression2

Example:

```
if (x<0)
        flag = 0;
else
        flag = 1;
```

The above statement can be written as

```
flag = (x<0) ? 0 : 1;
```

Math.h (header file)

- Most of the mathematical functions are placed in math.h header
- Some are specified in the stdlib.h header
- Some common mathematical functions:

Function Name	Description
exp(x)	returns *e* raised to the given power (e^x)
sqrt(x)	computes square root ($\sqrt{x}$)
cos(x)	computes cosine (*cos(x)*)
pow(x,y)	raises a number to the given power (x^y) [pow(x.y)]
sinh(x)	computes hyperbolic sine (*sinh(x)*)
erf(x)	error function
And so on…….	tan(x), abs(x), log10(x)….etc

- The outputs of the functions are of the **double format.**

Math.h header file

- Math Constants:

Constant Name	Description
M_E	The base of natural logarithms (e).
M_LOG2E	The base-2 logarithm of e.
M_PI	3.141593
M_SQRT2	The positive square root of 2.
M_SQRT1_2	The positive square root of 1/2.
And so on…….	

Practice Example

```
#include<stdio.h>
#include<math.h>
int main()
{
double pi;
pi=M_PI;  //sets pi = 3.1416
double sum;
sum=cos(pi);
//here in cos(x) , x is radian value, so input should be radian
printf("%lf",sum);
return 0;
}
```

output:
-1.000000

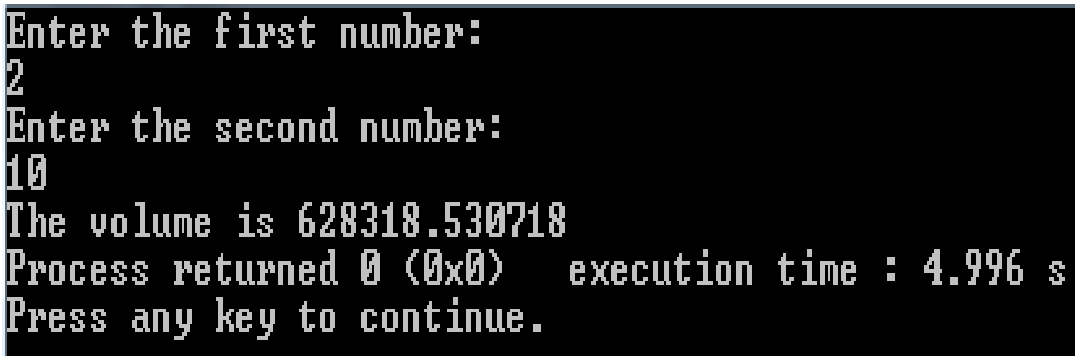
Class Performance 2

- Write a program that takes two numbers as input.
- Find the square root of the first number and the resulting output will be the radius of a cylinder.
- Raise the second input number to a power of 5. The resulting output will be the height of the cylinder.
- Find the volume of the cylinder by using the saved value of pi in the header file.

- **Remember to use the math.h file.**

ANSWER

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{   int a,b;
   double r,h,pi,V;
   pi= M_PI;
   printf(“Enter the first number: \n");        scanf("%d",&a);
   printf(“Enter the second number: \n");   scanf("%d",&b);
   r= sqrt(a);
   h= pow(b,5);
   V=pi*r*r*h;
   printf("The volume is %lf",V);
   return 0; }
```



The getch() function

The **getch( )** function reads a single character the instant it's typed without waiting for ENTER.

get means it gets something i.e. it's an input function
ch means it gets a character

The getche() function

The **getche( )** function also reads a single character the instant it's typed without waiting for ENTER **and also echoes it**.

get means it gets something i.e. it's an input function
ch means it gets a character
e means it echoes the character to the screen when you type it.

Example

```
#include <stdio.h>
void main(void)
{
 char test;
 printf("Type any character: ");
 test = getch();
 printf("\nThe character you typed was:
 %c",test);
}
```

```
Type any character:
The character you typed was: s
Process returned 0 (0x0)   execution time : 0.728 s
Press any key to continue.
```

Replace getch() with getche()

Conditional Statements

The *if* statement

General form:

```
if (condition)
{
    statement;
}
```

Conditions:

1. Using relational or conditional operators
2. Using logical operators

Multiple statements within *if*

General form:

```
if (condition)
{
 statement 1;
 statement 2;
 ----------;
 statement n;
}
```

Example of *if statement*

Write the following program with multiple statements

```
int i;
printf("Enter an Integer: ");
scanf("%d",&i);
if (i==1)
{
   printf("\n You typed 1");
   printf("\n End of statement");
}
printf("\n End of the program");
```

If you type 1, Output:
Enter an Integer: 1
You typed 1
End of Statement
End of the program

If you type any other no., except1 .Output:
Enter an Integer: 3
End of the program

General form

if-else

```
if (condition)
{
        statement 1;
        statement 2;
}

else
{
        statement 1;
        statement 2;
}
```

Note: else is optional

if-else if-else

```
if (condition)
{
        statement 1;
        statement 2;
}

else if (condition)
{
        statement 1;
        statement 2;
}

else
{
        statement 1;
        statement 2;
}
```

Example of *if-else* statement

Write the following program with multiple statements

```
int i;
printf("Enter an Integer: ");
scanf("%d",&i);
if (i==1)
{
   printf("\n You typed 1");
}
else
{
   printf("\n You did not type 1");
}
printf("\n End of the program");
```

Example of if- *else if*- else statement

```
int num;
printf("Enter an Integer: ");
scanf("%d",&num);
if (num < 0)
{
        printf("\n the number is less than zero");
}
else if(num == 0)
{
        printf("\n the number is equal to zero");
}
else
{
        printf("\n the number is greater than zero");
}
```

```
Enter an Integer: 12

 the number is greater than zero
Process returned 33 (0x21)   execution time : 2.503 s
Press any key to continue.
```

Nested *if-else* statements

General form

```
if (condition)
{
        statement;
}

else
{
       if (condition)
       {
         statement;
       }

       else
       {
        statement;
       }
}
```

Another form

```
if (condition)
{
        if (condition)
        {
                statement;
        }
        else
        {
                statement;
        }
}

else
{
    statement;

}
```

Class Performance 3

Write a C program to prepare an electricity bill.

No. of Units consumed	Amount of bill
Less than or equal to 100	200
Between 101 and 130	250
Between 131 and 150	275
Over 150	300

Take the number of units consumed as input and print the amount of total bill as output.

ANSWER

```
void main()
{           int x;
            printf("Enter the amount of bill: ");
            scanf("%d",&x);
if (x<=100)
{       printf("The amount of bill is 200");   }
else if(x>101 && x<130)
{       printf("The amount of bill is 250");   }
else if(x>131 && x<150)
{       printf("The amount of bill is 275");   }
else
{       printf("The amount of bill is 300");      }
}
```

Assignments:

1. Write a C program to find the smallest of 3 integers taken as input using nested if-else statement .

2. Write a C program to find the roots of a Quadratic Equation ax2+bx+c =0, that will take coefficients a, b, c as input and find the roots as output.. Use nested if-else statement.

Thank you

Everything has its beginning. But it doesn't start at "one."

-Metal Gear Solid 4