ME 172

Computer Programming Language Sessional

Lecture 2: Operators and Conditional Statements

Mantaka Taimullah

Lecturer

Md. Kamruzzaman

Assistant Professor

Dept of Mechanical Engineering, BUET

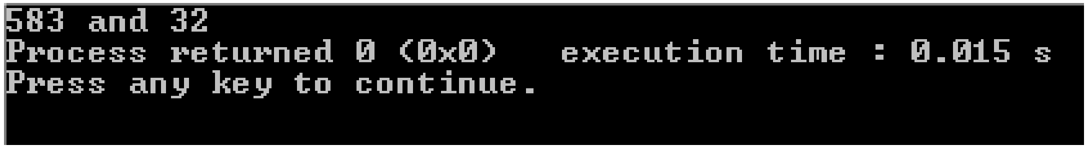
Performance Test 1

- 29765 apples are to be divided into 51 buyers. Write a C program that will calculate how many apples each buyer gets and how many apples are left and then display the results.

Time: 5 minutes!!

ANSWER

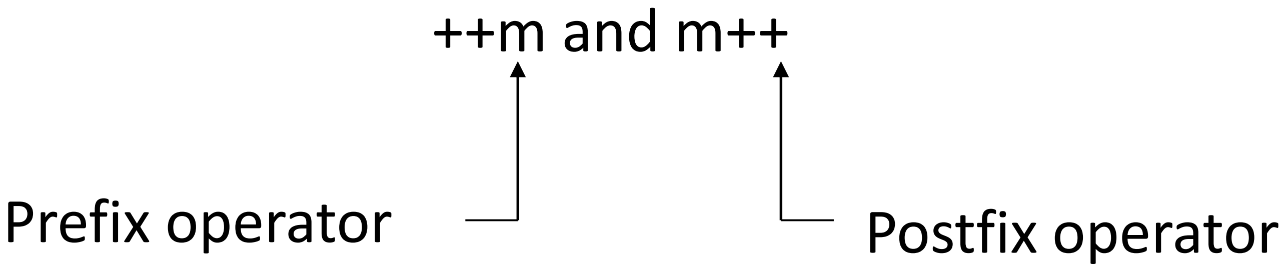
```
#include <stdio.h>

int main()
{
    int a,b,c,d,e,f;
    a=29765;
    b=51;
    c=29765/51;
    d=29765%51;
    printf("%d and %d",c,d);
    return 0;
}
```



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

Arithmetic operator(Contd...)

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



++m and m++

Prefix operator

Postfix operator

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").

Arithmetic 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

Arithmetic Operators(Contd...)

- **Write the following program:**

```
#include<stdio.h>

int main()
{
  int a=10,b=20,x;
  x=a*++b;
  printf("\n The value of x is: %d”,x);
  return 0;
}
```

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

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

Arithmetic Operator(Contd...)

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

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), fabs(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
```

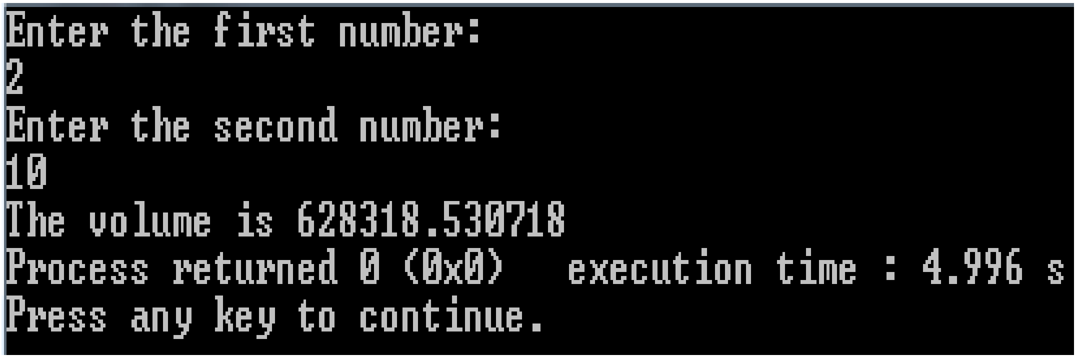
Class Performance 2

- 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;
 }
```



```
Enter the first number:
2
Enter the second number:
10
The volume is 628318.530718
Process returned 0 (0x0)   execution time : 4.996 s
Press any key to continue.
```

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

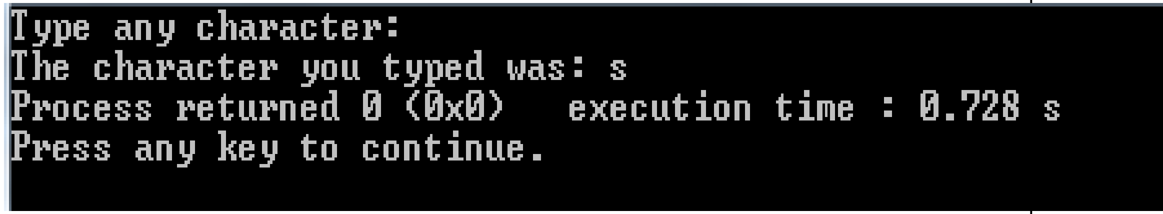
ch means it gets a character

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*

Program to display a number if user enters negative number

```
#include <stdio.h>
int main()
{
  int number;
  printf("Enter an integer: ");
  scanf("%d", &number);
  if (number < 0)
  {
     printf("You entered a negative number");
  }
 printf(“\nYou entered %d”, number);
 return 0;
}
```

```
Enter an integer: -3
You entered a negative number
You entered -3
```

```
Enter an integer: 5
You entered 5
```

Conditions

Comparison operators

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

- 1if 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.

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

Program to check whether an integer entered by the user is odd or even

```
#include <stdio.h>
int main()
{
 int number;
 printf("Enter an integer: ");
 scanf("%d",&number);
 if( number%2 == 0 )
        printf("%d is an even integer.", number);
 else
        printf("%d is an odd integer.", number);
 return 0;
}
```

```
Enter an integer: 7
7 is an odd integer.
```

```
Enter an integer: 18
18 is a even integer.
```

An alternate to if-else
The Conditional operator (? :)

General form:

Conditional expression? Expression1 : Expression2

Example

```
#include <stdio.h>
int main()
{
  int number;
  printf("Enter an integer: ");
  scanf("%d", &number);
  (number % 2 == 0) ? printf("%d is even.", number) : printf("%d is odd.", number);
  return 0;
}
```

Example of if- *else if*- else statement

Program to compare two user given number

```c
#include <stdio.h>
int main()
{
  int number1, number2;
  printf("Enter two integers: ");
  scanf("%d %d", &number1, &number2);
  if(number1 == number2)
  {
             printf(“The numbers are equal");
  }
  else if (number1 > number2)
  {
             printf(“The first number is larger");
  }
  else
  {
             printf(“The first number is smaller");
  }
  return 0;
}
```

```
Enter two integers: 12
23
The first number is smaller
```

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;

}
```

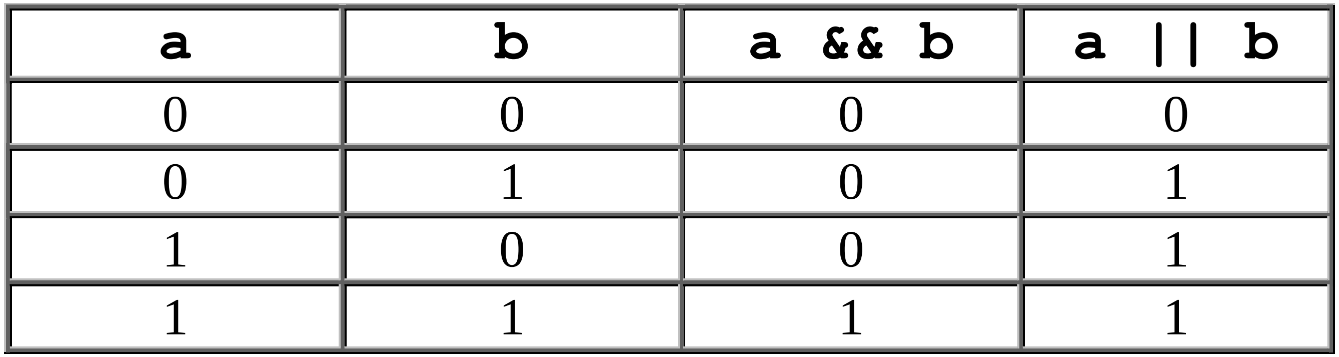
Example of Nested *if-else* statements

Program to compare two user given number

```
#include <stdio.h>
int main()
{
  int number;
  printf(“Enter an integer: “);
  scanf(“%d”, &number);

  if (number>0){
    if (number%2==0){
      printf(“You entered an even positive number”);
    }
    else{
      printf(“You entered an odd positive number”);
    }
  }
  else{
    printf(“The number you entered is not positive”);
  }
  return 0;
}
```

Use of more than one conditions in one *if* statement

Operator	Meaning	Example	
&&	And	if(num>0&&num%2==0)	Will be executed if both statements are true
||	Or	if(num>0||num%2==0)	Will be executed if any of the conditions are true



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

Exercise

Write a Program to check that whether a number is positive or negative.

Instructions:

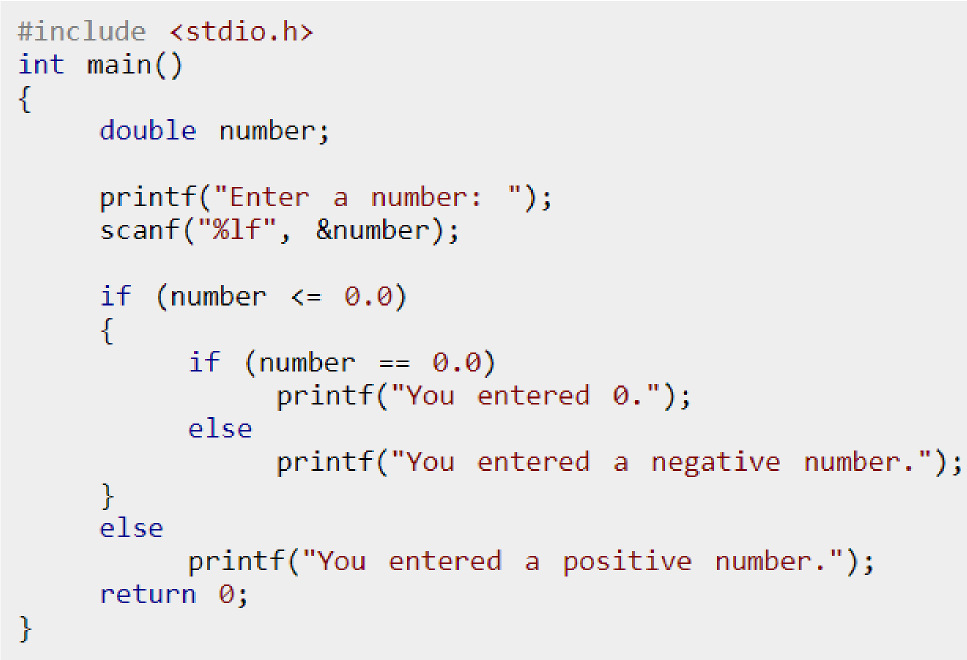
Number must be taken as input.

Number may be integer or float type.

You can use nested if-else or if-else if-else statement.

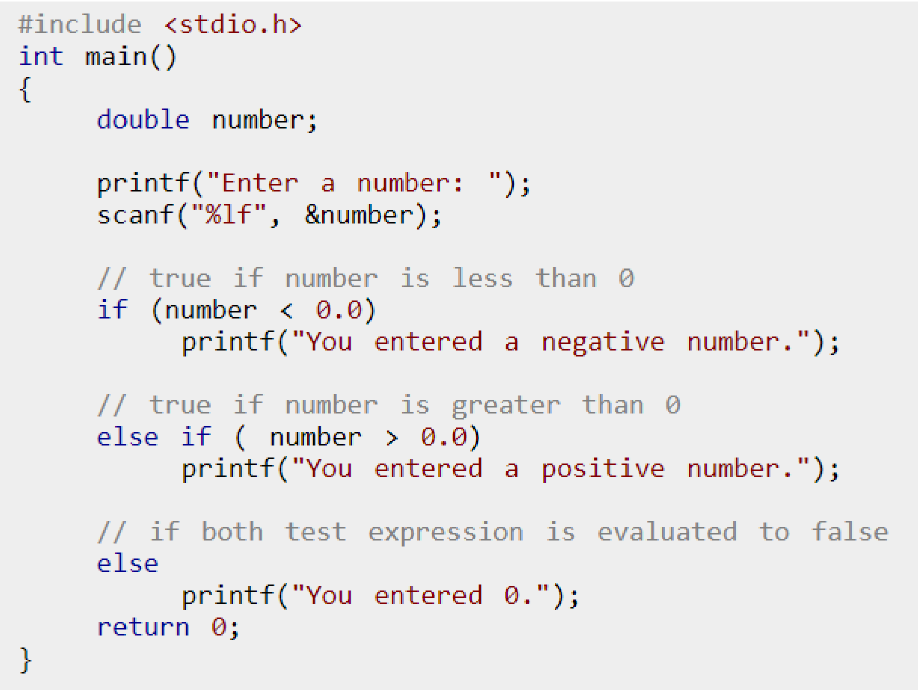
ANSWER

Nested if-else statement



```
#include <stdio.h>
int main()
{
    double number;

    printf("Enter a number: ");
    scanf("%lf", &number);

    if (number <= 0.0)
    {
        if (number == 0.0)
            printf("You entered 0.");
        else
            printf("You entered a negative number.");
    }
    else
        printf("You entered a positive number.");
    return 0;
}
```

if-else if-else statement



```
#include <stdio.h>
int main()
{
    double number;

    printf("Enter a number: ");
    scanf("%lf", &number);

    // true if number is less than 0
    if (number < 0.0)
        printf("You entered a negative number.");

    // true if number is greater than 0
    else if ( number > 0.0)
        printf("You entered a positive number.");

    // if both test expression is evaluated to false
    else
        printf("You entered 0.");
    return 0;
}
```

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 that whether a Year is leap or not.**

Thank you