C Programming

Recitation 2

March 2016

Arithmetic Operators

- Arithmetic operators used to perform arithmetic operations (That's the magic!)
- There are five basic arithmetic operators in C language

Operator	Meaning
+	Addition (Unary plus)
-	Subtraction (Unary minus)
*	Multiplication
/	Division
%	Modulo (Remainder of a division)

EXAMPLE

```
#include <stdio.h>
int main() {
        int a=9,b=4,c;
        c=a+b;
        printf("a+b=%d\n",c);
        c=a−b;
        printf("a−b=%d\n",c);
        c=a*b;
        printf("a*b=%d\n",c);
        c=a/b;
        printf("a/b=%d\n",c);
        c=a%b;
        printf("Remainder when a divided by b=%d\n",c);
        return 0;
}
```

Increment and Decrement Operators

- ++ and -- are called increment and decrement operators respectively
- ++ adds 1 to operand
- -- subtracts 1 to operand

```
int a=5;
a++;   //a becomes 6
a--;   //a becomes 5
++a;   //a becomes 6
--a;   //a becomes 5
```

++/-- AS PREFIX

- The syntax is ++*var* (--*var*)
- Increment (Decrement) the value of *var*
- Then return it

```
#include <stdio.h>
int main() {
    int a=b=0;
    printf("a: %d\n", ++a);
    /*
    a = a+1;
    printf("a: %d\n", a);
    */
    printf("b: %d\n", --b);
    return 0;
}
```

++/-- As Postfix

- The syntax is *var*++ (*var*--)
- Return the value of *var*
- Then, increase (decrease) the value it
- Notice the differences between this and previous slides.

```
#include <stdio.h>
int main() {
   int a=b=0;
   printf("a: %d\n", a++);
   /*
   printf("a: %d\n", a);
   a = a+1;
   */
   printf("b: %d\n", b--);
   return 0;
}
```

Assignment Operators

- The most common one is "="
- "=" assigns *r-value* to *l-value*

```
int a=0, b=1, c=4;
a = 1; // 1 is assigned to a
b = c; // value of c is assigned to b
0 = c; // Error! l-value should be a variable that can be ↩
    examined, altered. Not a constant!
```

Other Assignment Operators

- There are five more assignment operators

Operator	Example	Same As
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b

Nested Assignments

- You can mix arithmetic and assignment operators in a single line

```
#include <stdio.h>
int main() {
   int i=1, j=2, k=3;
   int res;
   i += j = k;
   res = i--j------k;
   printf("1) i: %d j: %d k: %d\n", i, j, k);
   printf("2) res: %d i: %d j: %d k: %d\n", res, i, j, k);
   printf("3) i: %d j: %d k: %d\n", i, j, k);
}
```

Logical Operators

- &&, ||, and ! are the three logical operators

expr1	*expr2*	&&	||	!*expr1*	!*expr2*
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

EXAMPLE

- Open logical.c file that contains following lines:

```
#include <stdio.h>
int main(){
   int expr1 = 0, expr2 = 0;
   printf("and: %d or: %d not(expr1): %d not(expr2): %d\n",
   expr1&&expr2, expr1||expr2, !expr1, !expr2);
   expr1 = 0; expr2 = 1;
   printf("and: %d or: %d not(expr1): %d not(expr2): %d\n",
   expr1&&expr2, expr1||expr2, !expr1, !expr2);
   expr1 = 1; expr2 = 0;
   printf("and: %d or: %d not(expr1): %d not(expr2): %d\n",
   expr1&&expr2, expr1||expr2, !expr1, !expr2);
   expr1 = 1; expr2 = 1;
   printf("and: %d or: %d not(expr1): %d not(expr2): %d\n",
   expr1&&expr2, expr1||expr2, !expr1, !expr2);
}
```

Details

- For logical operations
  - 1 means *true*
  - 0 means *false*
  - Any number, except 0, means *true*

```
int expr1 = 62, expr2 = -140, expr3;
expr3 = expr1 && expr2; //true
printf("expr1 && expr2: %d\n", expr3);
expr3 = expr1 || expr2; //true
printf("expr1 || expr2: %d\n", expr3);
expr3  = !expr1; //false
printf("!expr1: %d\n", expr3);
```

Relational Operators

- We will use relational operators to compare data (variables, constants etc.)
- C supports six relational operators

==	is equal to
!=	is not equal to
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to

EXAMPLE

```c
/* Relational Operators */
#include <stdio.h>
int main() {
   int a = 111, b = 140;
   printf("a == b: %d", a == b);
   printf("a != b: %d", a != b);
   printf("a < b: %d", a < b);
   printf("a <= b: %d", a <= b);
   printf("a > b: %d", a > b);
   printf("a >= b: %d", a >= b);
   return 0;
}
```

Conditional Operators

- Three operands
- Works from left to right
- *condition ? True_parts : False_parts*

```
/* Conditional Operators */
#include <stdio.h>
int main(){
    char feb;
    int days;
    printf("Enter 1 if the year is leap year otherwise enter 0: ");
    scanf("%c",&feb);
    days=(feb=='1')?29:28;
    /*If test condition is true, days will be equal to 29. */
    /*If test condition is false, days will be equal to 28. */
    printf("Number of days in February = %d",days);
    return 0;
}
```

If – Else Statements

- You have seen if statements in Python
- Logic is the same, syntax is different

```
if (expr)
   statement_1 /* do this if expr is non-zero (true) */
else
   statement_2 /*do this if expr is zero (false) */
```

- The **else** clause is optional
- *expr* may include:
  - Relational operators
  - Logical operators
  - Arithmetical operators (not commonly)
  - Combination of them

If – Else If – Else Statements

- The **elif** token in Python is replaced with **else if** clause in C

```
if (expr_1)
   statement_1 /* do this if expr_1 is non-zero (true) */
else if (expr_2)
   statement_2 /* do this if expr_2 is non-zero (true) */
else
   statement_3 /*do this if all expressions are zero (false) */
```

- If *expr_1* is correct, following (related) "else if" and "else" conditions will not be checked.
- Control mechanism checks conditions from top to bottom.
- Last **else** is always attached to last **if**

If – Else Example

```
/* If – Else Example 1*/
#include <stdio.h>
int main()
{
   int a, b;
   printf("Please enter a and b respectively");
   scanf("%d %d", &a, &b);
   if (!(a || b) || (a && b))
      printf("Result is 1");
   else
      printf("Result is 0");
}
```

If – Else Example

```
/* If – Else Example 2*/
#include <stdio.h>
int main(){
   int numb1, numb2;
   printf("Enter two integers to check\n");
   scanf("%d %d", &numb1, &numb2);
   if (numb1 == numb2) //checking whether two integers are ↩
      equal.
       printf("Result: %d = %d", numb1, numb2);
   else if (numb1 > numb2) //checking whether numb1 is greater ↩
      than numb2.
          printf("Result: %d > %d", numb1, numb2);
   else
          printf("Result: %d > %d", numb2, numb1);
   return 0;
}
```

If – Else Example

```
/* If – Else Example 3*/
#include <stdio.h>
int main(){
   int numb1, numb2;
   printf("Enter two integers to check\n");
   scanf("%d %d", &numb1, &numb2);
   if (numb1 == numb2) //checking whether two integers are ↩
      equal.
       printf("Result: %d = %d", numb1, numb2);
   else
       if (numb1 > numb2) //checking whether numb1 is greater ↩
      than numb2.
           printf("Result: %d > %d", numb1, numb2);
       else
           printf("Result: %d > %d", numb2, numb1);
   return 0;
}
```

Common Errors

- Do not put ; or : after
  - **if**(expr) / **else if**(expr) / **else**
  - after } parentheses
- Be careful about operands
  - **if**(a > b) vs **if**(a >> b)
  - **if**(a == 5) vs **if**(a = 5)
  - etc.

Math Library

- Performs common mathematical calculations
- To call the desired function
  - Include the header in your source code
  - Add the compilation flag **-lm** to your compile command
- You can reach all available functions in this library at
  - `http://www.cplusplus.com/reference/cmath`
- You may need to use **cos()**, **sin()**, **log()**, **log10()**, **pow()**, **sqrt()** functions in lab exam

Math Library

Function	Description	Example
cos(x)	trigonometric cosine of x (x in radians)	cos(0.0) is 1.0
sin(x)	trigonometric sine of x (x in radians)	sin(0.0) is 0.0
log(x)	natural logarithm of x (base e)	log(2.718282) is 1.0
log(x)	logarithm of x (base 10)	log(10.0) is 1.0
pow(x, y)	x raised to power y (x^y)	pow(2,7) is 128.0
sqrt(x)	square-root of x	sqrt(9.0) is 3.0

Usage

```
#include <stdio.h>
#include <math.h>
int main(){
   int a, b;
   printf("Enter a negative number:");
   scanf("%d", &a);
   if (a < 0){
      b = pow(a, 2);
      printf("Square of %d is %d\n", a, b);
      printf("Square-root of %d is %d\n", b, sqrt(b));
      if (sqrt(b) == sqrt(pow(a, 2)))
         printf("You can combine the functions!\n");
   }
   else
      printf("Input is invalid. Program will be terminated!\n");
   return 0;
}
```

How to Compile

- Compile and run as follows:

```
gcc mathLib.c -Wall -ansi -pedantic-errors -o mathLib -lm
./mathLib
```

- Notice the **-lm** flag that we have added to compile our code with "math.h" library

Lab Demo

- Let's use what we have learned today and complete simple task in Moodle.

Evaluation

- This time evaluation.sh and most of the inputs are not available to you to demonstrate a lab exam environment
- You can check your grade only on Moodle
- Therefore, if you prefer to work locally, please regularly upload your file and check your grade.