CL1002 – Programming Fundamentals Lab

Lab # 05

Data Types & Expressions

Instructor: Muhammad Saood Sarwar

Email: saood.sarwar@nu.edu.pk

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar

Data Types

C supports the following basic data types:

int: integer, a whole number.

float: floating point, a number with a fractional part.

char: single character.

Type	Size (Bytes)	Format Specifier
int	At least 2, usually 4	%d
float	4	%f
char	1	%c

Example 1

```c
#include <stdio.h>
int main()
{
   int i = 23;
   float f = 3.14;
   char c = 'H';
   printf("int: %d\n", i);
   printf("float: %f\n", f);
   printf("char: %c\n", c);
   return 0;
}
```

Output

int: 23

float: 3.140000

char: H

C Increment and Decrement Operators

C programming has two operators increment ++ and decrement -- to change the value of an operand by 1.

Increment ++ increases the value by 1 whereas decrement -- decreases the value by 1. These two operators are unary operators, meaning they only operate on a single operand.

Example 2

```
#include <stdio.h>
int main()
{
   int a = 10, b = 100;
   float c = 10.5, d = 100.5;

   printf("++a = %d \n", ++a);
   printf("--b = %d \n", --b);
   printf("++c = %f \n", ++c);
   printf("--d = %f \n", --d);
   return 0;
}
```

Output

++a = 11

--b = 99

++c = 11.500000

--d = 99.500000

The sizeof operator

The sizeof is a unary operator that returns the size of data i-e how much memory it takes.

Example 3

```
#include <stdio.h>
int main()
{
printf("Size of int  =%ld bytes\n",sizeof(int));
printf("Size of float=%ld bytes\n",sizeof(float));
printf("Size of char =%ld byte\n",sizeof(char));

return 0;
}
```

Output

Size of int = 4 bytes

Size of float=4 bytes

Size of char =1 byte

Expression

An expression is a combination of operators, constants and variables.

Examples:

x+y

x+y-1

x+1

Operator Precedence

There can be more than one operator in an expression.

To evaluate these types of expressions there is a rule of precedence. It guides the order in which these operations are carried out.

For example, multiplication has higher precedence than subtraction.

But we can change this order using parentheses () as it has higher precedence than multiplication.

C evaluates a numeric expression based on operator precedence.

The + and – are equal in precedence, as are *, /, and %.

The *, /, and % are performed first in order from left to right and then + and -, also in order from left to right. You can change the order of operations by using parentheses ( ) to indicate which operations are to be performed first.

For example, the result of 5 + 3 * 2 is 11, whereas the result of (5 + 3) * 2 is 16.

Example 4

```c
#include <stdio.h>
int main()
{
   int a = 6;
   int b = 4;
   int c = 2;
   int result;

   result = a - b + c;         //  4
   printf("Result = %d\n",result);

   result = a + b / c;         //  8
   printf("Result = %d\n",result);

   result = (a + b) / c;         //  5
   printf("Result = %d\n",result);
   return 0;
}
```

Output:

Result = 4

Result = 8

Result = 5

Assignment Operators

Assignment operators are used to assign values to variables.

int a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5

For example:

```
int x = 3;
x = x + 1;     // x is now 4
```

To shorten this type of assignment statement, C offers the += assignment operator. The statement above can be written as

```
x += 1;   // x = x + 1
```

Many C operators have a corresponding assignment operator. The program below demonstrates the arithmetic assignment operators:

Example 5

```c
#include <stdio.h>
int main()
{
   int a = 5, c;

   c = a;       // c is 5
   printf("c = %d\n", c);
   c += a;      // c is 10
   printf("c = %d\n", c);
   c -= a;      // c is 5
   printf("c = %d\n", c);
   c *= a;      // c is 25
   printf("c = %d\n", c);
   c /= a;      // c is 5
   printf("c = %d\n", c);
   c %= a;      // c = 0
   printf("c = %d\n", c);

   return 0;
}
```

Output

c = 5

c = 10

c = 5

c = 25

c = 5

c = 0

C Input

In C programming, `scanf()` is one of the commonly used function to take input from the user. The `scanf()` function reads formatted input from the standard input such as keyboards.

Example 6

```c
#include <stdio.h>
int main()
{
   int testInteger;
   printf("Enter an integer: ");
   scanf("%d", &testInteger);
   printf("Number = %d",testInteger);
   return 0;
}
```

Output:

Enter an integer: 3

Number = 3

Here, we have used `%d` format specifier inside the `scanf()` function to take int input from the user. When the user enters an integer, it is stored in the `testInteger` variable.

Example 7

C program to compute $(a^2 + b^2)$ The formula of $(a^2 + b^2)$ is given below.

$(a^2 + b^2) = a^2 + b^2 + 2ab$

```c
#include <stdio.h>

int main()

{

   int a,b,ans;

   a=3;

   b=2;

   ans = (a*a) + (b*b) + (2 * a * b);

   printf("Result =  %d\n",ans);

   return 0;

}
```

Output

Result = 25