Programming Fundamentals Lab

Lab # 04

Variables and Arithmetic Operators

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2022

Department of Computer Science,

National University of Computer and Emerging Sciences FAST
Peshawar Campus

Contents

Variable

- A named memory location where data is stored is called variable.
- A quantity whose value may change during execution of the program is called variable. It is represented by a symbol or name.
- **Variable** is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*.
- It is a combination of "vary + able" that means its value can be changed. ✦ int data=10 // Here data is variable.

Rules for defining variables

● A variable can have alphabets, digits, and underscore.

● A variable name can start with the alphabet, and underscore only. It can't start with a digit.

● No whitespace is allowed within the variable name.

● A variable name must not be any reserved word or keyword, e.g. int, float, etc.

Valid variable names:

int a;

int _ab;

int a30;

Invalid variable names:

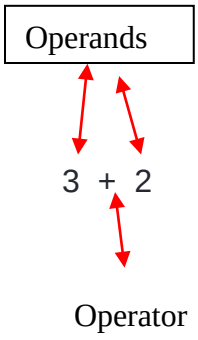
int 2;

int a b;

int long;

Operators

Operators are special symbols that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.



Here, `+` is the operator that performs addition. `2` and `3` are the operands and `5` is the output of the operation.

```
int sum1 = 100 + 50;         // 150 (100 + 50)
int sum2 = sum1 + 250;       // 400 (150 + 250)
int sum3 = sum2 + sum2;      // 800 (400 + 400)
```

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

Operator	Meaning	Example
+	Add two operands	x + y
-	Subtract right operand from the left	x - y
*	Multiply two operands	x * y
/	Divide left operand by the right one.	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)

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

Expression

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

Examples:

x+y

x+y-1

x+1

Precedence of Operators

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.

Precedence of arithmetic operators from top to bottom:

*

/

%

+

-

Example Programs

Sum of two Numbers

```
#include<stdio.h>
int main()
{
   int a=2;
   int b=3;
   int sum=a+b;
   printf("\nSum: %d", sum);
   return 0;
}
```

Output:

Sum: 5

Calculator:

```
#include<stdio.h>
int main()
{
   int a=2;
   int b=3;
   int sum=a+b;
   int diff=a-b;
   int product=a*b;
   int division=a/b;
   int mod=a%b;
   printf("a = %d  b = %d", a, b);
   printf("\nSum: %d", sum);
   printf("\nDiff: %d", diff);
   printf("\nProduct: %d", product);
   printf("\nDivision: %d", division);
   printf("\nModulus: %d", mod);
   return 0;
}
```

Output:

a = 2 b = 3

Sum: 5

Diff: -1

Product: 6

Division: 0

Modulus: 2

Area of a Square:

```
#include<stdio.h>
int main()
{
   int side=5;
   int area=side*side;
   printf("Area of the square is: %d", area);
   return 0;
}
```

Output:

Area of the square is: 25

Feet to Inches Converter:

```
#include<stdio.h>
int main()
{
   int feet=5;
   int inches=feet*12;
   printf("%d Feet in inches: %d", feet, inches);
   return 0;
}
```

Output:

5 Feet in inches: 60