

# **Chapter 3:**

# **OPERATORS,**

# **EXPRESSIONS**

# **AND I/O FUNCTIONS**



- An **operator** is a function which is applied to values to give a result.
- You should be familiar with operators such as +, -, /. Arithmetic operators are the most common
- Operators and values are combined to form expressions.

# I. Arithmetic Operators

## Arithmetic Operators

The symbols of the arithmetic operators are:

Operation	Operator	Comment	Value of Sum before	Value of sum after
Multiply	*	<code>sum = sum * 2;</code>	4	8
Divide	/	<code>sum = sum / 2;</code>	4	2
Addition	+	<code>sum = sum + 2;</code>	4	6
Subtraction	-	<code>sum = sum - 2;</code>	4	2
Increment	++	<code>++sum;</code>	4	5
Decrement	--	<code>--sum;</code>	4	3
Modulus	%	<code>sum = sum % 3;</code>	4	1



## 2. The Relational Operators

These allow the comparison of two or more variables.

### **Operator Meaning:**

**==** equal to

**!=** not equal

**<** less than

**<=** less than or equal to

**>** greater than

**>=** greater than or equal to



# 3. Logical Operators (AND, NOT, OR)

Combining more than one condition

- These allow the testing of more than one condition as part of selection statements.

The symbols are

- Logical AND    &&
- Logical and requires all conditions to evaluate as TRUE (non-zero).
- Logical OR    ||
- Logical or will be executed if any ONE of the conditions is TRUE (non-zero).
- Logical NOT    !
- Logical not negates (changes from TRUE to FALSE, vsvs) a condition.



## 4. Conditional operator

- This conditional expression operator takes 3 operators.  
The 2 symbols used to denote this operator are ? and the :. The first operand is placed before ?, the second one between the ? and the : and the third after:

The general format is,

Conditional ? expression 1 : expression 2

- If the result of condition is TRUE ,expression 1 is evaluated and the result of the evaluation becomes the result of the operation.

If the condition is FALSE, the expression 2 is evaluated and its result becomes the result of the operation

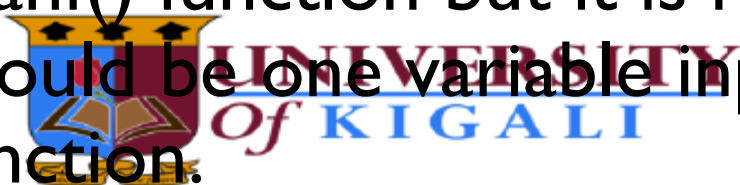


- Examples:
- 1)
- $s = (x < 0) ? -1 : x * x;$
- if  $x$  is less than zero then  $s = -1$
- if  $x$  is greater than zero then  $s = x * x$

# READ AND WRITE FUNCTIONS IN C

## The input I/O manipulator: Scanf

- **Scanf(“Format String”, &variable);**
- Here **format string** is used to define which type of data it is taking as input this format string can be %c for character, %d for integer variable and %f for float variable.
- **& sign** is an operator that tells the compiler the address of the variable where we want to store the value.
- One can take multiple input of variable with single scanf() function but it is recommended that there should be one variable input with one scanf() function.





# Example

*/\* program which introduces keyboard input \*/*

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
int number;
```

```
printf("Type in a number \n");
```

```
scanf("%d", &number);
```

```
printf("The number you typed was %d\n", number);
```

```
getchar();
```

```
}
```



# Area of the circle

```
#include <stdio.h>
```

```
access*/
```

```
main()
```

```
Heading*/
```

```
{
```

```
float radius, area;
```

```
Declaration*/
```

```
printf("radius=");
```

```
prompt*/
```

```
scanf("%f", &radius);
```

```
area=3.14*radius*radius;
```

```
Statement */
```

```
printf("Area of circle= %f",area);
```

```
/*getchar();*/
```

```
}
```

```
/* Library file
```

```
/* Function
```

```
/* Variable
```

```
/* Output statement
```

```
/* Input Statement*/
```

```
/* Assignment
```

```
/* output Statement */
```



**UNIVERSITY**  
*Of* **KIGALI**

- 2)
- `#include<stdio.h>`
- `main()`
- `{`
- `int a,b,c;`
- `printf("Enter the numbers:");`
- `scanf("%d %d",&a,&b);`
- `c=(a>b)?a:b;`
- `printf("The biggest number is %d", c);`
- `getchar();`
- `}`

3) `y=(x>5)?3:4;`

- 4) Evaluate the following expression, where `a=4,b=5;`
- `least_value=(a<b)?a:b;`



# Summary

- Every C program contains a function `main()` that controls execution of the program
- Every C program statement must end in a semicolon
- Use `printf` to display strings and variable values.
- Always code the preprocessor directive `# include <stdio.h>` in all your programs
- A variable is a named location in computer memory that stores a particular type of data.
- A variable declaration must begin with the variable's data type.
- The integer are `int`, `unsigned`, `long`, `unsigned long`, `short`, and `unsigned short`.
- Use `scanf` to obtain a datum from the input stream(the keyboard) and place in to a variable.
- The main arithmetic operators are `+`, `-`, `*`, `/`, and `%`.
- The arithmetic operators follow the usual precedence rules.
- Use `const` to declare defined constants.
- A statement is a complete direction instructing the computer to carry out some task
- A group of one or more statement enclosed within braces is called a **Block**.



- **Question2:** Write a **C** program to read the two input integer values and manipulate them using arithmetic operators.
- **Question 3:** write a program to use the conditional operator with two values.
- **Question 4:** What is the Out put ?  $2 = 3 ? 4 : 5$
- **Question 5:** A program to enter two values from the keyboard and then find maximum using conditional operator
- **Question6:** Write a program to use the conditional operator with two values.
- **Question7:** Write a program to use the conditional operator with two statements.

