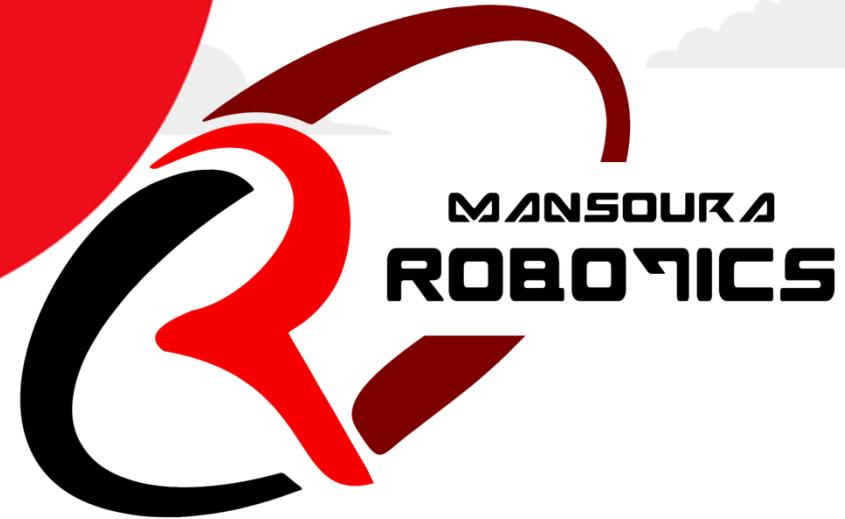


Conditional Statements in C

Lecture 2



01

**Relational and logical
operators in C**

02

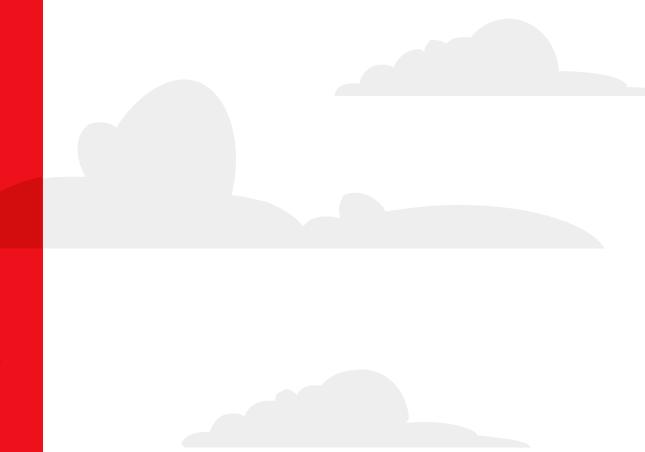
If statement in C

03

Switch statement in C

01

Relational and logical operators in C



Remember C operators

Arithmetic	Uni	++			--				
	Bi	+		-	*	/	%		
Bit wise	&		~	^	>>	<<			
Assignment	=	+=	-=	*=	/=	%=	+=		
	&=	=	^=	>>=	<<=				
Relational	>	<	>=	<=	==	!=			
Logical	&&					!			
Other	Size of operator			sizeof()					
	Ternary operator			? :	;				
	Address operator			&	(will be discussed later)				
	Dereference			*	(will be discussed later)				
	Subscriptor			[]	(will be discussed later)				

True and False in C

True



Any number not equals to zero

False



0

1 is true

1000 is true

-4 is true

Any number except 0 is true

Note, if there is a statement in c that returns true, the compiler is free to choose the value of the true with the rule that it must be any number except 0.

Relational operators in C

This operators are used to check the relation between new values and return either true or false.

```
int x = 10;  
int y = 5;
```

1- Check Equality

example `x == y /* checks if x equals to y
 this statement will return false */`

2- Check Not Equality

example `x != y /* checks if x is not equal to y
 this statement will return true */`

3- Check More Than

example `x > y /* check if x is more than y
 this statement will return true */`

4- Check More Than or equal

example `x >= y /* check if x is more than or equals to y
 this statement will return true */`



Relational operators in C

This operators are used to check the relation between new values and return either true or false.

```
int x = 10;  
int y = 5;
```

5- Check Less Than

example

```
x < y /* check if x is less than y  
this statement will return false */
```

6- Check Less Than or equal

example

```
x <= y /* check if x is less than or equals to y  
this statement will return false */
```

Note, if you tried to print false value, the value that will be printed is 0

Note, if you tried to print true value, the value that will be printed is not zero and chosen by the compiler, in most cases it will be 1 or 255



Logical operators in C

These operators are used to apply logical operation between two values, each value will be considered either false if it is 0 or true if it is not 0.

1- Logical And

example

```
int x = 3 && 0; /* 3 is true and 0 is false  
true AND false is false  
then x now equals 0 */
```

2- Logical OR

example

```
int x = 3 || 0; /* 3 is true and 0 is false  
true OR false is true  
if the compiler consider true is 255  
then x now equals 255 */
```

3- Logical Not

example

```
int x = !5 ; /* 5 is true  
NOT true is false  
then x now equals 0 */
```



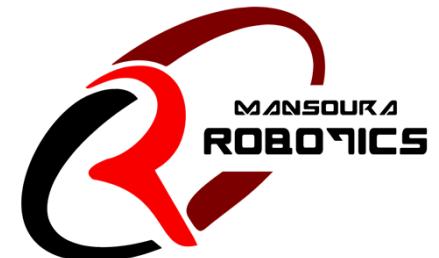
LAB1

Expected Output

```
False value in GCC = 0  
True  value in GCC = 1
```

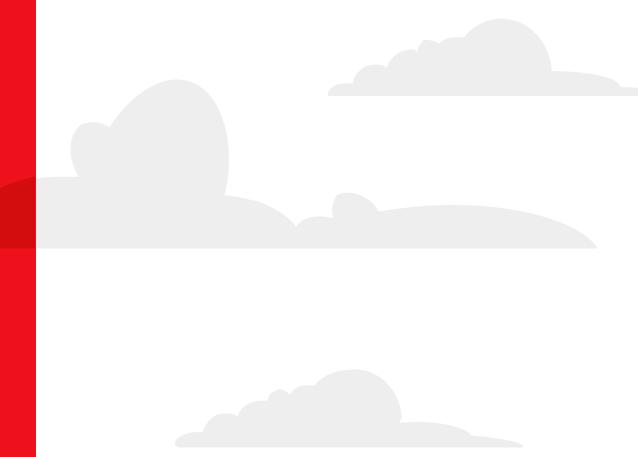
Write a C code to print the false value and the true value on GCC.

*Time to
Code*



02

If statement in C



Conditional statements in C

Conditional statements are used to execute some code under certain conditions.

C defines 2 different conditional statements.

1- if Statement

2- switch Statement

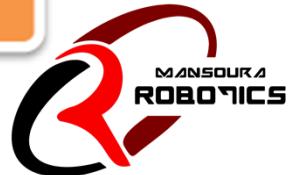
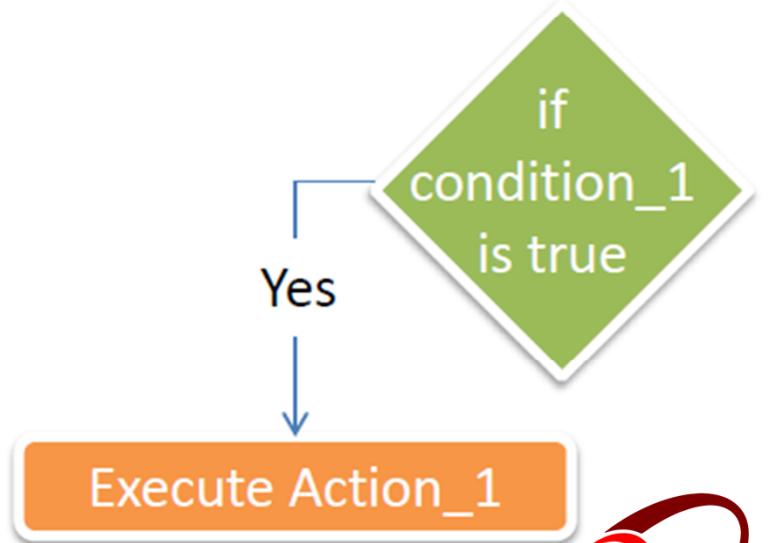


If statement in C

Case_1 : Only if condition

Syntax

```
if ( condition_1 )
{
    Action_1
}
```



LAB2

Expected Output

```
Please Enter Your working hours: 50  
Your Salary is 2500
```

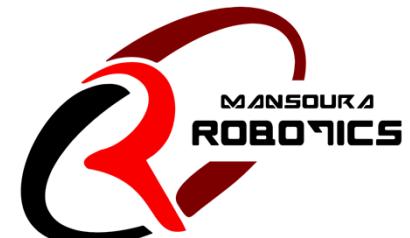
```
Please Enter Your working hours: 20  
Your Salary is 900
```

Write a C code to calculate employee salary in a week based on the his working hours, hour rate is 50.

The program will ask the user to enter the working hours, then it will print his salary.

But if the working hours are less than 40 hours, a 10% deduction will be applied.

*Time to
Code*



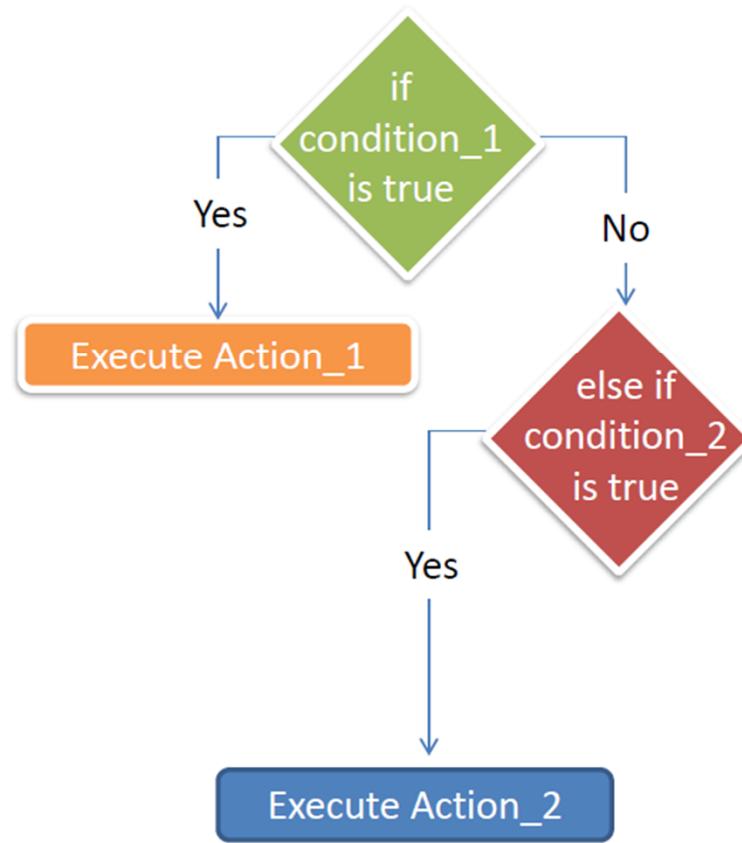
If statement in C

Case_2 : if, else if statement

Syntax

```
if ( condition_1 )
{
    Action_1
}
```

```
else if ( condition_2 )
{
    Action_2
}
```



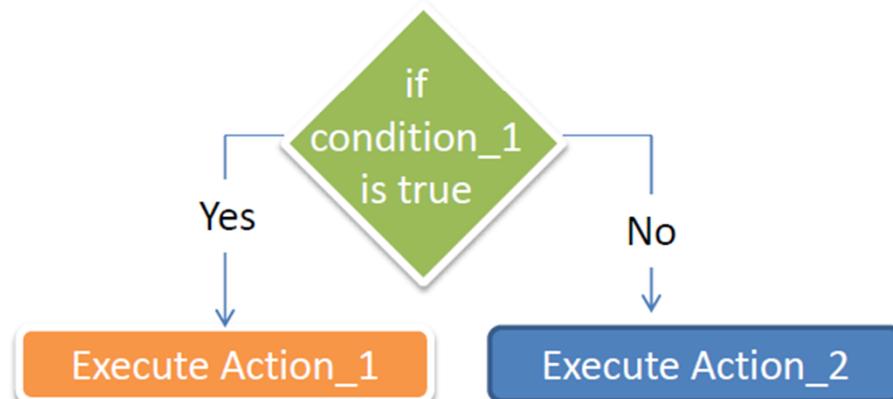
If statement in C

Case_3 : if, else statement

Syntax

```
if ( condition )
{
    Action_1
}
```

```
else
{
    Action_2
}
```



LAB3

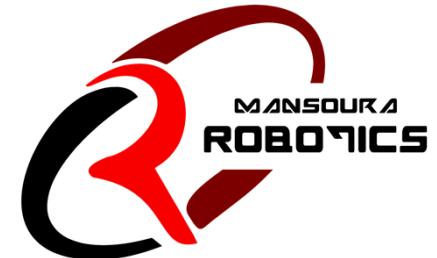
Expected Output

Write a C code that ask the user to enter a number and check if it is Even or Odd number

Please enter number: 6
Number is Even

Please enter number: 7
Number is Odd

Time to
Code



If statement in C

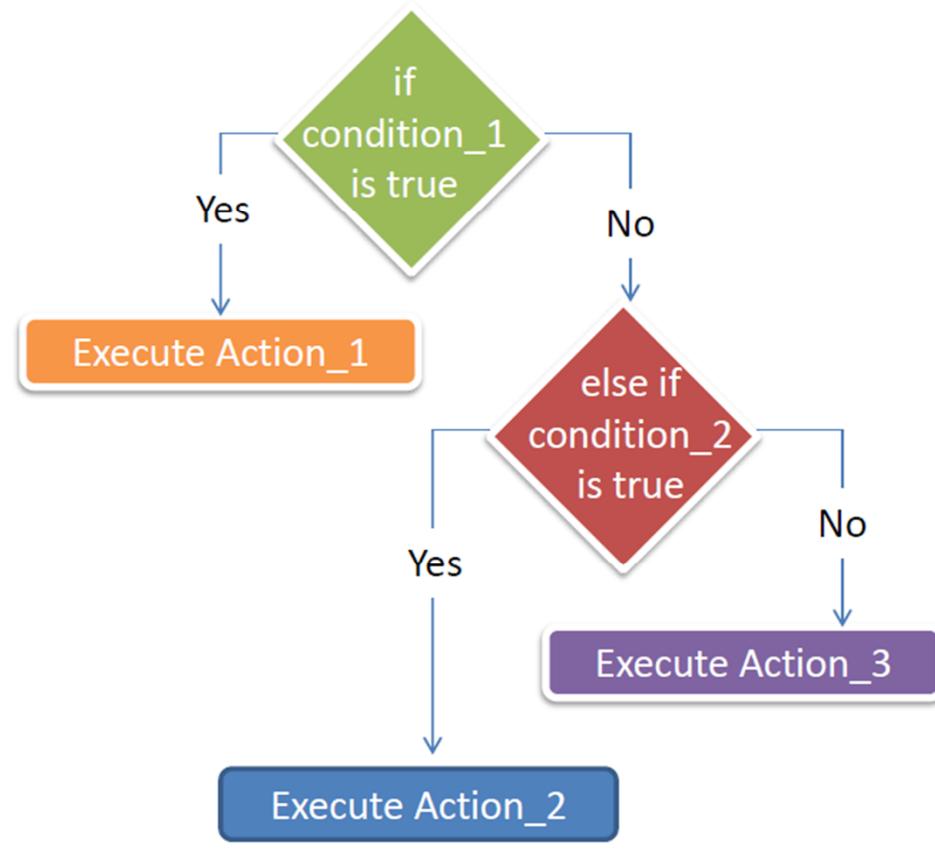
Case_4 : if, else if, else condition

Syntax

```
if ( condition_1 )  
{  
    Action_1  
}
```

```
else if ( condition_2 )  
{  
    Action_2  
}
```

```
else  
{  
    Action_3  
}
```



Lab4

Expected Output

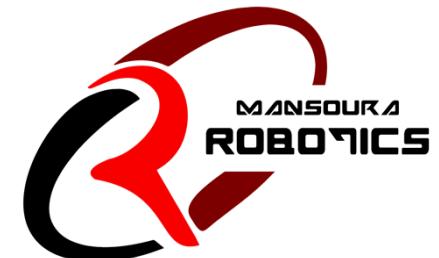
Write a C code to ask the user to enter his grade and the program will print his rating.

0 <= grade < 50 -----> Failed
50 <= grade < 65 -----> Normal
65 <= grade < 75 -----> Good
75 <= grade < 85 -----> Very Good
85 <= grade -----> Excellent

Please enter number: 63
Your rating is Normal

Please enter number: 92
Your rating is Excellent

Please enter number: 45
Your rating is failed



Time to
Code

If statement in C

General Rules:

- 1- **else if** statement is optional, you may have no else if, you may have one, you may have more, no limit.
- 2- **else** statement is optional, you can have only one else statement.
- 3- No code is allowed to be written between if and else if or else.
- 4- Nested if is allowed.
- 5- if the condition in if statement is a combination between many conditions, use round brackets () with each condition to avoid precedence issues.
- 6- In case of only one action only should be taken in if statement, you may not use { }, but it is always preferred to use { } even if only one action is required.



Question 1

What will be the output of the following code ... ?

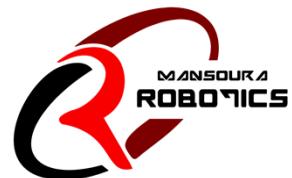
- a- Ahmed
- b- Youssef
- c- compilation error
- d- 30

```
#include <stdio.h>

void main(void)
{
    int x = 10;
    if ( x == 10 )
    {
        printf ("Ahmed");
    }

    x = 30;

    else
    {
        printf ("Youssef");
    }
}
```



Question 1

What will be the output of the following code ... ?

a- Ahmed

b- Youssef

c- compilation error

d- 30

Reason:

No code is allowed between
if and else

```
#include <stdio.h>

void main(void)
{
    int x = 10;
    if ( x == 10 )
    {
        printf ("Ahmed");
    }
    x = 30;
    else
    {
        printf ("Youssef");
    }
}
```



Question 2

What will be the output of the following code ... ?

- a- Ahmed
- b- Youssef
- c- compilation error
- d- Ahmed
Samir

```
#include <stdio.h>

void main(void)
{
    int x = 10;
    if ( x == 10 )
        printf ("Ahmed\n");

    else
        printf ("Youssef\n");
        printf ("Samir");

}
```



Question 2

What will be the output of the following code ... ?

- a- Ahmed
- b- Youssef
- c- compilation error
- d- Ahmed
Samir

Reason:

No { } with else statement,
then only one statement is
corresponding to else.

```
#include <stdio.h>

void main(void)
{
    int x = 10;
    if ( x == 10 )
        printf ("Ahmed\n");
    else
        printf ("Youssef\n");
        printf ("Samir");
}
```



03

Switch statement in C

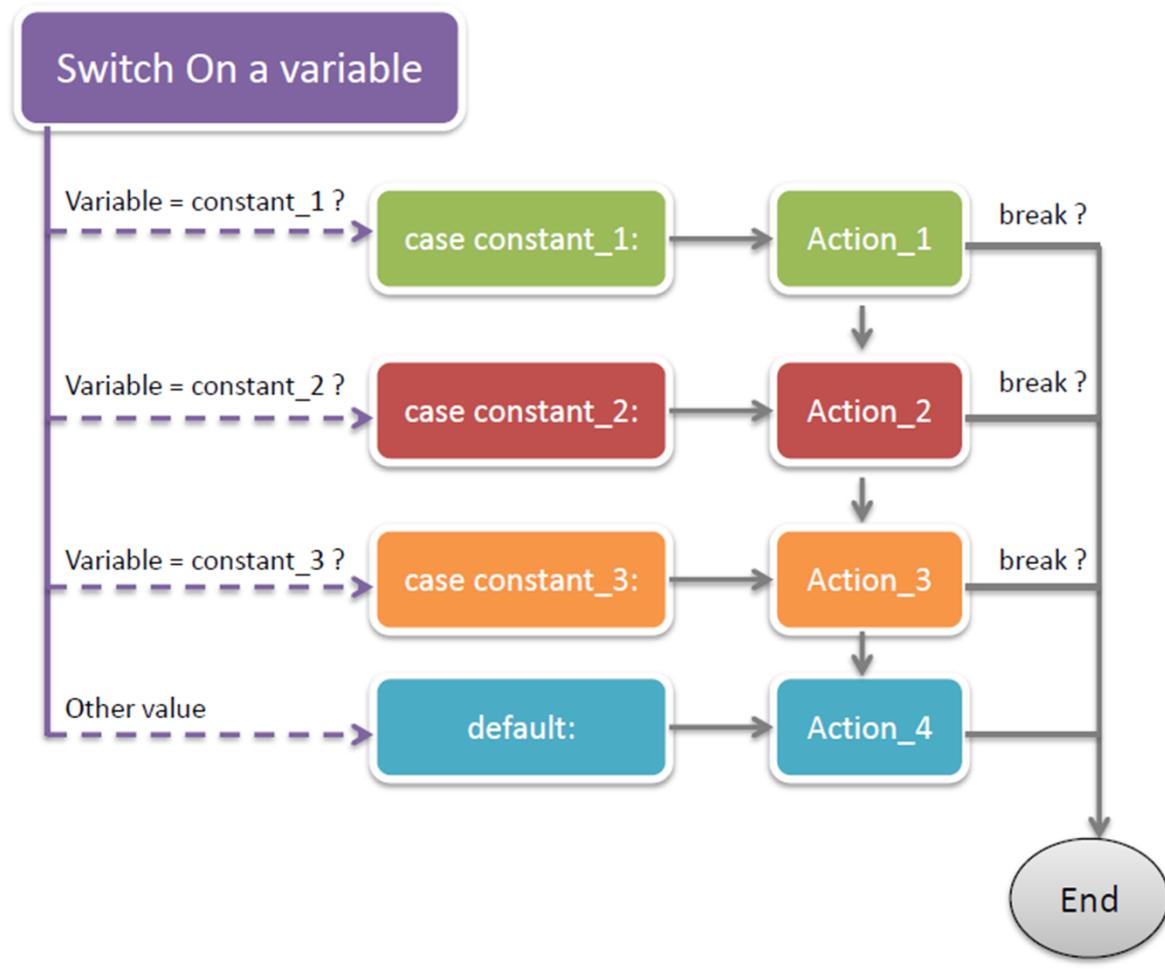


Switch Statement in C

Switch statement is a control statement that allows us to choose only one choice among many choices.

It compares the variable value with the values present in the different cases. Then it executes that block of code which matches the case value.

If there is no match, then default block is executed



LAB5

Login System:

Write a C code that ask the user to enter his ID and then the program will print his name.

Available IDs are:

1234-> Ahmed

5678 -> Youssef

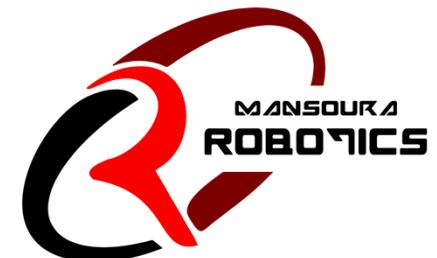
1145 -> Mina

Any other number, the program will print **Wrong ID**

Expected Output

Please Enter Your ID: 1234
Welcome Ahmed

Please Enter Your ID: 8897
Wrong ID



Time to
Code

Switch Statement Rules

- 1.case constant **must be unique**
- 2.case constant **can't be a variable**
- 3.case constant must be **integral value**
- 4.Only one default is allowed
- 5.default label is Optional
- 6.default can be placed **anywhere in the switch**
- 7.break Statement ends the switch
- 8.if the break statement is not exist, the all following code will be executed
until the end of the switch or until it finds a break statement without checking the case constant
- 9.Nesting (switch within switch) **is allowed.**



Thank you!

Do you have any questions?

Assignment 1

Write a C code that ask the user to enter 10 numbers, then ask him to enter another number to search on it in the 10 numbers and print its location in case it is found.

In case the number is not found, it will print number no exist

Expected Output

```
Enter Number 1: 5
Enter Number 2: 6
Enter Number 3: 8
Enter Number 4: 9
Enter Number 5: 11
Enter Number 6: 14
Enter Number 7: 34
Enter Number 8: 58
Enter Number 9: 12
Enter Number 10: 6
Enter the value to search: 12
Value is exist at element number 9
```



Assignment 2

Write a C code that ask the user to enter his ID, if the ID is valid it will ask the user to enter his password, if the password is correct the program will print the user name, if the password is incorrect the program will print ***Incorrect Password***.

In case of not existing ID, the program will print ***Incorrect ID***



Assignment 3

Write a code that will ask the user to enter 3 numbers, the program will print the maximum number of them.

Expected Output

```
Enter number 1: 5
Enter number 2: 3
Enter number 3: 9
Maximum number is 9
```

