Control Statements: Conditional Statements

- Introduction, conditional execution (if, if-else, nested if), and selection (switch), unconditional types (break, continue, goto).

What are Control flow statements?

- A program is set of instructions.
- On default these instructions are sequentially or linearly executed.
- *Instructions that break up the flow of execution of the program enable program to conditionally execute particular blocks of code, employ decision making, looping and branching statements.*
- *These are the statements which breaks the sequential execution of the program based on some condition.*

CONDITIONAL EXECUTION

Types of Conditional statements/decision making statements/ Control construct

- There are 5 Conditional statementsin C
  - ✓ Simple if control construct / One way selection statement
  - ✓ if else control construct / Two way selection statement
  - ✓ nested if else control construct
  - ✓ else if ladder or cascaded if else construct / Multi way selection statement
  - ✓ switch control construct / Multi-way statement

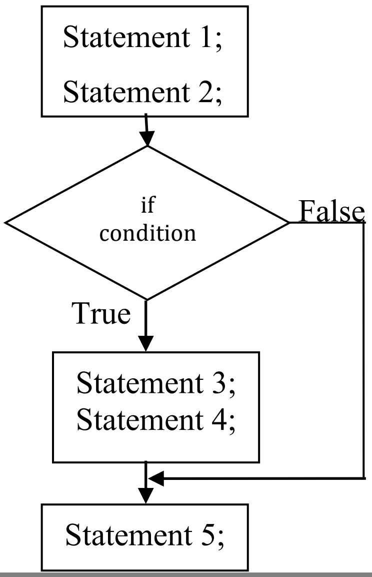
if statement (Simple if/ One way selection)

- An if statement is a single selection statement.
- It is used to execute a set of statements if the condition is true, if the condition is false, it skips executing those set of statements. Hence it is called one way selection.

Syntax:

```
Statement 1;
Statement 2;
if(condition)
{
      Statement 3;
      Statement 4;
}
      Statement 5;
```

Flowchart:



Explanation

- The keyword if must be followed by an expression and expression must be enclosed within parentheses.
- First statement1 is executed followed by statement2.
- Then Condition is checked
  - ✓ if false - control directly jumps to statement5 ignoring statement3 and 4.
  - ✓ if true - control goes to statement3 , statement4 and automatically goes to statement5.

An Example which illustrates if statement: ***To print given no is an even no.***

Algorithm	**Flowchart**	**Program**

To Check even number
Step 1: Start
Step 2: Read N
Step 3: if(n % 2 == 0)
{
Print “even no”
}
Step 4: Stop
	Draw Flowchart By Yourself.	#include<stdio.h>
		void main()
		{
		int n;
		clrscr();
		printf(“ Enter the number\n”)
		scanf(“%d”,&n);
		    if(n%2==0)
		      {
		      printf(“Even no”);
		      }
		getch();
		}

Note: Similarly write Algorithm, Flowchart and C Program for the following

- ❖ To print given no is an odd no. **Logic: if (n!=0)**
- ❖ To print given no is a positive no.**Logic: if(n>0)**
- ❖ To print given no is a negative no. **Logic: if(n<0)**

Disadvantage

- ✓ If one action has to be performed when the condition is true and another action has to be performed when the condition is false thenif-statement is not recommended. This disadvantage is overcome using two- way decision/selection statement called “ if-else statement”.

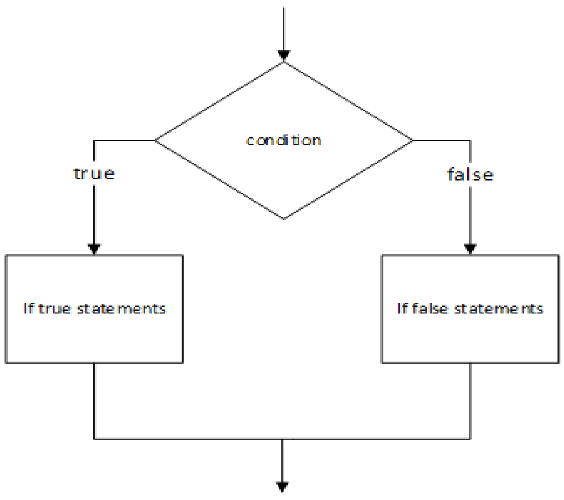
if – else statement (two way selection)

- It is used to execute a set of statements if the condition is true, and another set of statements if the condition is false. Hence it is called two way selections.

Syntax:

```
if (condition)
{
  // do this if condition is true
  // if true statements
}
else
{
  // do this is condition is false
  // if false statements
}
```

Flowchart:

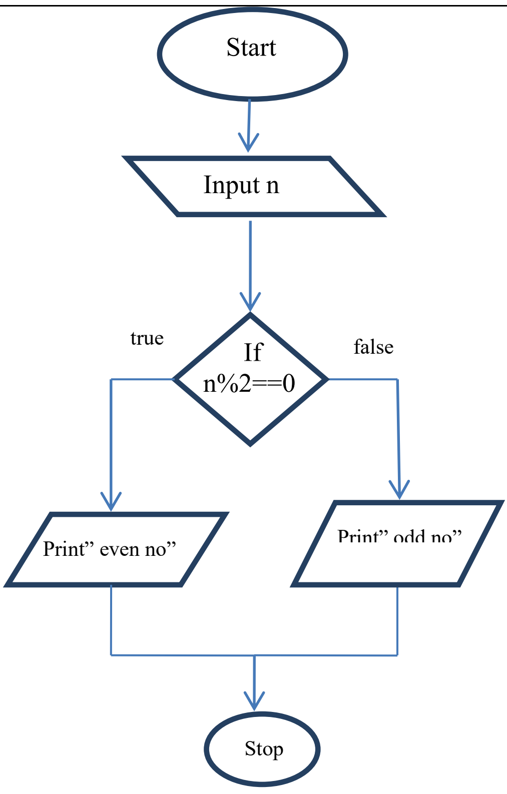


Explanation

- The keyword if and else must be followed by an expression and expression must be enclosed within parentheses.
- First statement1 is executed followed by statement2.
- Then Condition is checked
  - ✓ **If true**- control goes to if part wherestatement3 , statement4 are executed and automatically goes to statement7.
  - ✓ **If false** -control goes to else part where statement5, statement6are executed and automatically goes to statement7.
- In if-else either true part i.e., if part is executed or false part i.e., else part is executed based in the condition or test expression.

An Example which illustrates if-else statement: To print given no is an even no or odd no.

Algorithm	Flowchart	Program

Algorithm: Check even number or odd no
S1: Start
S2: Read N
S3: if(n % 2 == 0)
{
Print "even no"
}
else
{
Print "odd no"
}
S4: Stop
	 Start
	Input n
	true
	If n%2==0
	false
	Print" even no"
	Print" odd no"
	Stop
		#include<stdio.h>
		void main()
		{
		int n;
		clrscr();
		printf(" Enter the number\n")
		scanf("%d",&n);
		
		if(n%2==0)
		
		printf("Even no");
		else
		printf("odd no");
		
		getch();
		}

Note: Similarly write Algorithm, Flowchart and C Program for the following

- ❖ To check given integer no is a positive no or negative no.

  Logic: if(n>0) its positive else negative.

- ❖ To find largest of 2 no's .

  Logic: Let a and b be 2 no's if(a>b) *a is greater* else *b is greater*

- ❖ To check given no is even or odd

  Logic: Let n be a no's if(n%2==0) *n is even* else *n is odd.*

Nested if else

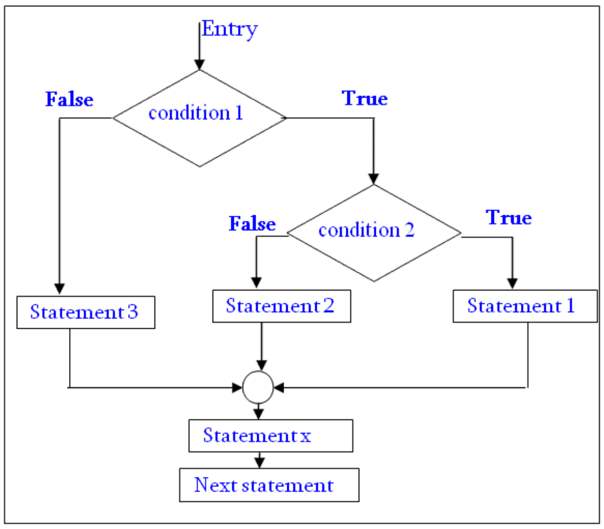
- It is used to execute one set of statements out of many set of statements depending upon the outcome of the conditions.
- It consists of if else control constructs with in another if or else control constructs and hence the name is nested if else.

Syntax:

```
if(condition-1)
        {
          if (condition-2)
                   Statement1;
          else
                   Statement2;
        }
else
         {
                Statement3;
         }
      Statement4;
```

s

Flowchart:



Explanation

- The keyword if and else must be followed by an expression and expression must be enclosed within parentheses.
- Then Condition is checked
  - ✓ Ifcondition-1 is true then again condition-2 is checked if both are true then Statement1 is executed.
  - ✓ Ifcondition-1 is true but condition-2 is false means Statement2 is executed.
  - ✓ If condition-1 itself is false control goes to Statement3 and automatically goes to Statement4.

Advantage:

- When an action has to be performed based on many decisions involving various types of expressions and variables then nested if statement is used.

Disadvantage:

- Difficult to understand and modify. As depth of nesting increases, the readability of the program decreases.

An Example which illustrates if-else statement: To find biggest of three numbers.

Algorithm	Flowchart	Program

Algorithm: To find largest of 3 no
S1: Start
S2: Read a,b,c
S3: if(a>b)
{
if(a>c)
{
Print “a largest”
}
else
{
Print “c largest”
}
}
else
{
if(b>c)
{
Print “b largest”
}
else
{
Print “c largest”
}
}
S4: Stop
	Draw the flowchart by yourself.	#include<stdio.h>
		void main()
		{
		inta,b,c;
		clrscr();
		printf(“ Enter the 3 numbers\n”);
		scanf(“%d%d%d”,&a,&b,&c);
		if(a>b)
		{
		if(a>c)
		{
		printf(“a largest”);
		}
		else
		{
		printf(“c largest”);
		}
		else
		{
		if(b>c)
		{
		printf(“b largest”);
		}
		else
		{
		printf(“c largest”);
		}
		}
		getch();
		}

Note: Similarly write Algorithm, Flowchart and C Program for the following

- ❖ To check given integer no is a positive no or negative no or zero.
- ❖ To find the greatest of three numbers.
- ❖ Magic number program.