Problem Solving Through programming in C
Course Code: ONL1001

DECISION MAKING & BRANCHING (Conditional branching)

Topic: IF Statement

Ms. SHUBHRA DWIVEDI
School - SCOPE
VIT-AP Amaravati

Introduction

- “Decision making and branching” is one of the most important concepts of computer programming.

- Programs should be able to make logical (true/false) decisions based on the condition provided.

- Every program has one or few problems to solve. In order to solve those particular problems important decisions have to be made depending on the nature of the problems.

- Generally C program execute it’s statements sequentially. But in order to solve problems we may have some situations where we have to change the order of executing the statements based on whether some conditions have met or not. So controlling the execution of statements based on certain condition or decision is called decision making and branching.

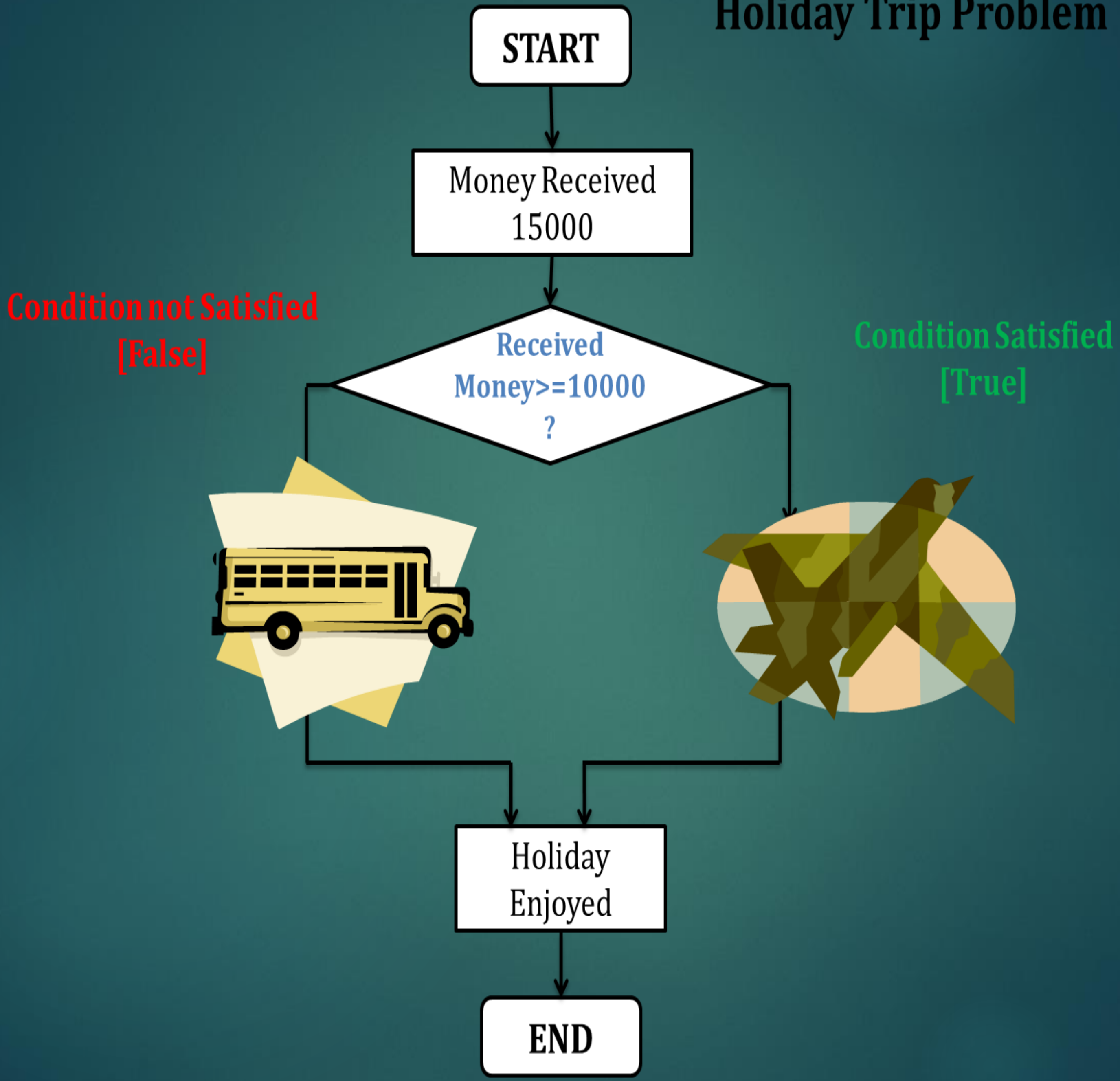
Holiday Trip Problem

Consider the fact that you and some of your friends have planed to go out for a holiday trip after the 1st year examination, 2020.

You have also decided that if you have got received money 10000 taka or more from your parent then your will go out for a foreign trip. Otherwise, if the allotted money is less than 10000 then you will go out for a country side trip.

Now you are supposed to design a program to solve this problem.

Holiday Trip Problem



START

Money Received 15000

Received Money>=10000 ?

Condition not Satisfied [False]

Condition Satisfied [True]

Holiday Enjoyed

END

Decision Making & Branching in C

- Instruction of a programs are executed either
  - Sequential manner
  - Branching
- C language possesses decision making and branching capabilities by supporting the following statements:
  - if statement
  - switch statement
  - conditional operator
  - goto statement
- These statements are knows as decision making statements. They are also called control statements as the control the flow of execution.

IF Statement

- The if statement is a powerful statement for decision making and is used to control the flow of execution of statements. It is basically a two-way decision making statement and is used in conjunction with an expression. It takes the following structure:

```
if (test-condition)
```

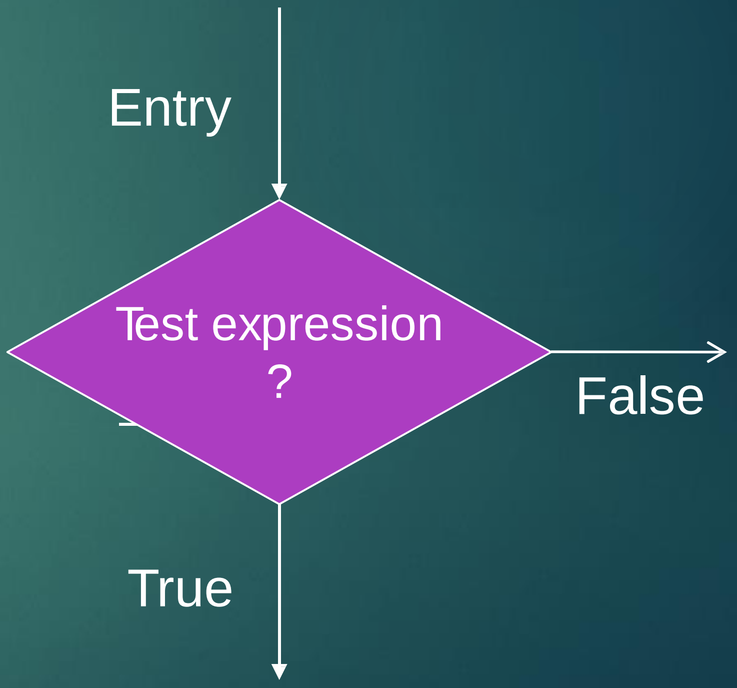
- It allows the computer to evaluate the expression first and then depending on whether the value of the expression or condition is true or false, it transfer the control to a particular statement. This point of program has two paths to follow, one for the true condition and the other for the false condition.

if statement

•It is used to control flow of execution of statement.

•It is two-way decision statement and is used in conjuction with an expression

```
Syntax-    if  (condition)
        {     statement 1;
                ………………..
         }
```



Ex: if (age is more than 55)

Person is retired

DIFFERENT FORMS OF IF STATEMENT

simple if

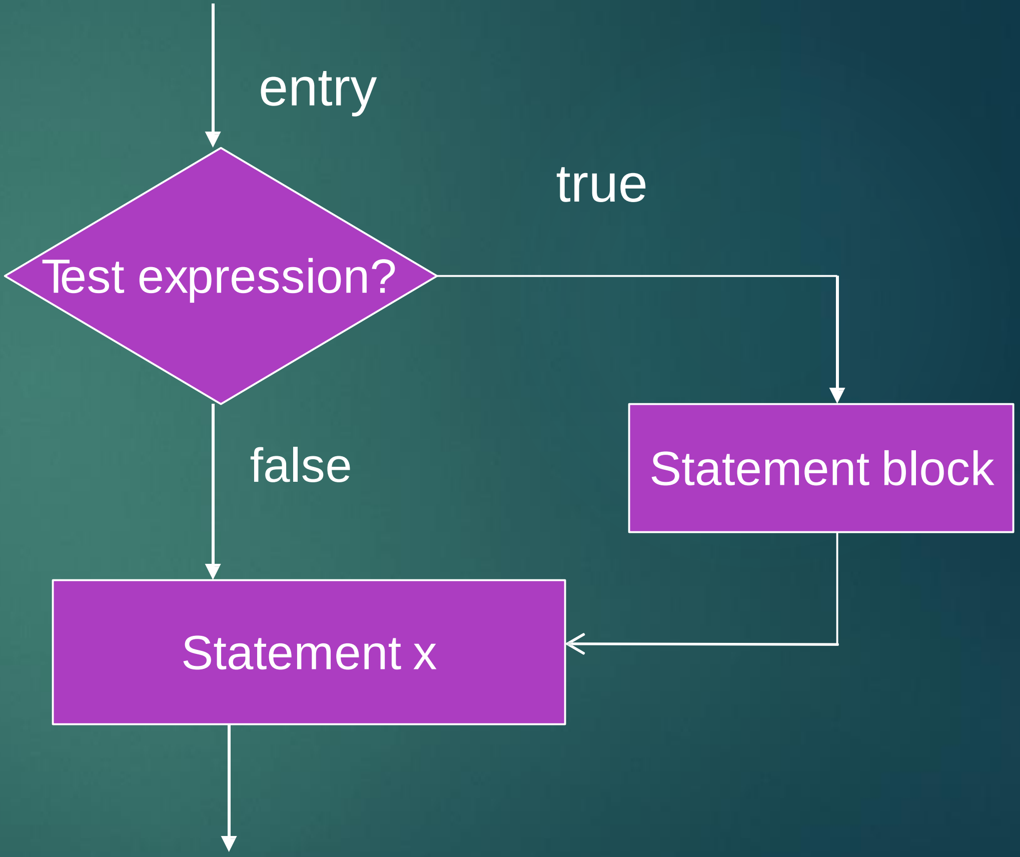
if-else

nested if-else

else if ladder

Simple if statement

```
Syntax-  if(Condition)
 {
         statement block;
   }
statement x ;
Ex: if ( category == sports)
   {
         marks=marks+bonus;
    }
printf ( “%f”,marks);
```



For example:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int age;
clrscr();
printf("Enter value of age");
scanf("%d",&age);
if(age>18)
{
printf("he is eligiable for voting");
}
printf("%d",age);
getch();
}
```

In Another Words, It Can Also Be Said As Single Blocked Conditional Statements In Which Only One Part Is Mentioned. That Is Known As TRUE Part.

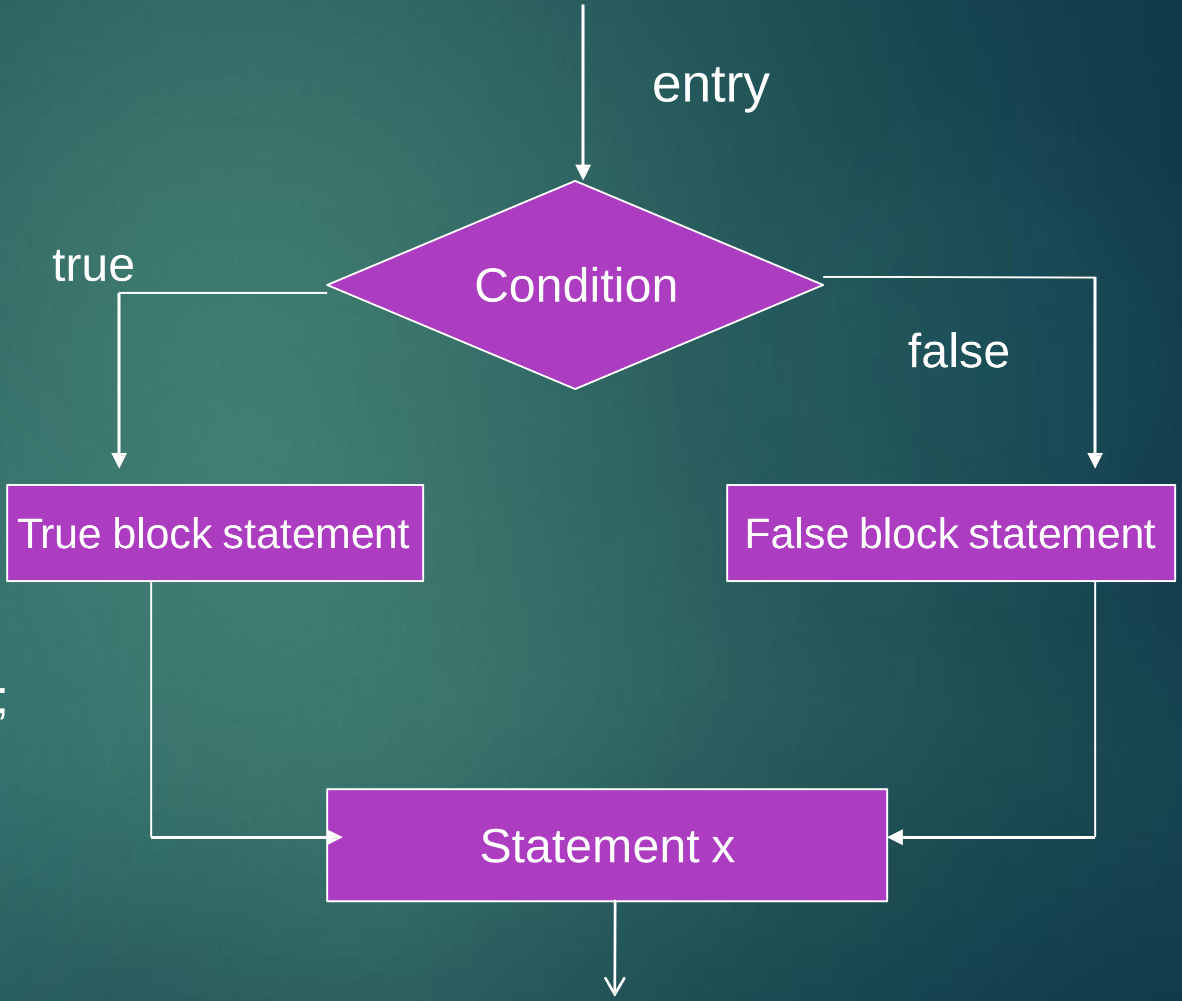
```
/*Any Number is input through the keyboard. write a If program.*/

#include<stdio.h>
#include<conio.h>
void main()
{
   int n;
   n=1;
   clrscr(); //clrscr() is the function of #include< conio.h > header file
which will clear previous output of program

   printf("Enter the Number");
   scanf("%d",&n);
   if(n>0)
   {
      printf("It is If Statement");
   }
   getch();
}
```

if else Statement

- If the test condition is true then the true block statements, immediately following the if statements are executed;
- Otherwise the false block statements are executed.
- In short either true-block or false-block of statements will be executed, not both.
- But in both cases the control is transferred subsequently to the statement-x as it is an independent (not controlled by the if else statement) statement.
- It is also called two way conditional branching.

The if…else statement

```
Syntax-   if( Condition)
    {
       true block statement;
     }
  else
   {
       false block statement;
    }
  statement x;
```



Continue....

```
Ex-:  if (code== 1)
boy=boy+ 1;
if ( code== 2)
 girl=girl+1;

 if (code==1)
  boy=boy+1;
 else
 girl=girl+1;
```

```
/*Any Number is input through the keyboard. write a program to
find out whether It is and Odd Number or Even Number.*/

#include< stdio.h >
#include< conio.h >
void main()
{
    int n;
    n=1;
    clrscr();
    printf("Enter the Number");
    scanf("%d",&n);
    if(n%2==0)
    {
        printf("This is Even Number");
    }
    else
    {
        printf("This is Odd Number");
    }
    getch();
}
```

```
output-  Enter an integer

              46

         This is  even number
```

Nested if else Statement

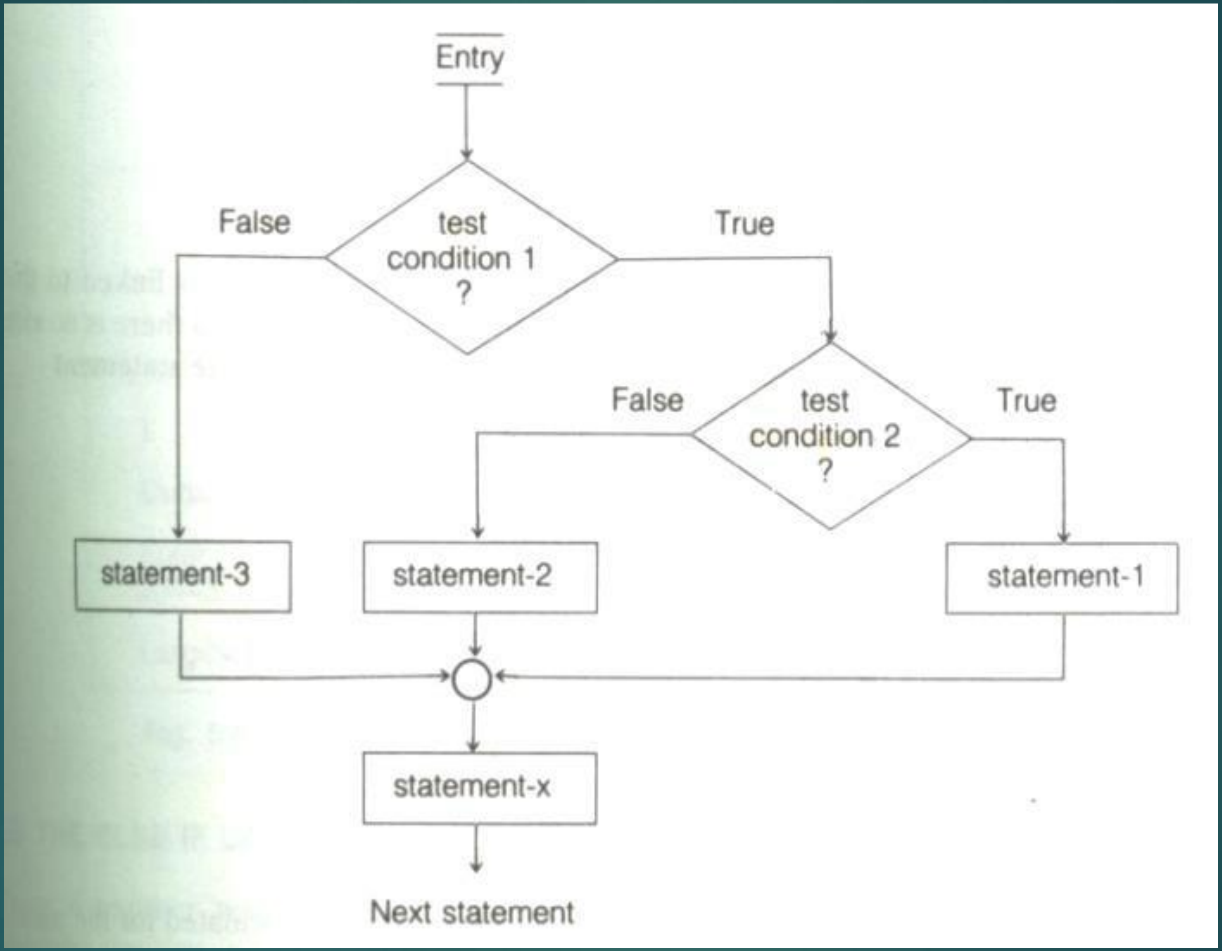
- Using “if…else statement” within another “if…else statement” is called ‘nested if statement’. “Nested if statements” is mainly used to test multiple conditions.
- It is called nested conditional branching.
- They can be structured using following syntax:

Syntax-

```
if( test condition 1 )
    {
    if (test condition 2)
       {
       statement 1;
       }
    else
      {
       statement 2;
       }
    }
else
   {
    statement 3;
   }
statement x   ;
```

Nested if else

NESTING OF IF…ELSE STATEMENT



```
/*If the ages of Ram, sham and Ajay are input through the keyboard,
write a program to determine the youngest of the three*/

#include< stdio.h >
#include< conio.h >
void main()
{
  int ram,sham,ajay;
  clrscr();
  printf("Enter the Three Ages of Ram,Sham and Ajay\n");
  scanf("%d%d%d",&ram,&sham,&ajay);
  if(ram < sham)
  {
    if(ram < ajay)
    {
      printf("Ram is Youngest");
    }
    else
    {
      printf("Ajay is Youngest");
    }
  }
else
 {
   if(sham < ajay)
   {
     printf("Sham is Youngest");
   }
   else
   {
     printf("Ajay is Youngest");
   }
 }
 getch();
}
```

```
Output is as :
Enter the three Ages of Ram,Sham and
Ajay
14
17
19
Ram is Youngest
```

Else if ladder Statement

- The word ladder means the staircase. As the name implies this statement is used to choose right way/paths among multiple paths.

- There is another way of putting if conditions together when multiway decisions are involved.

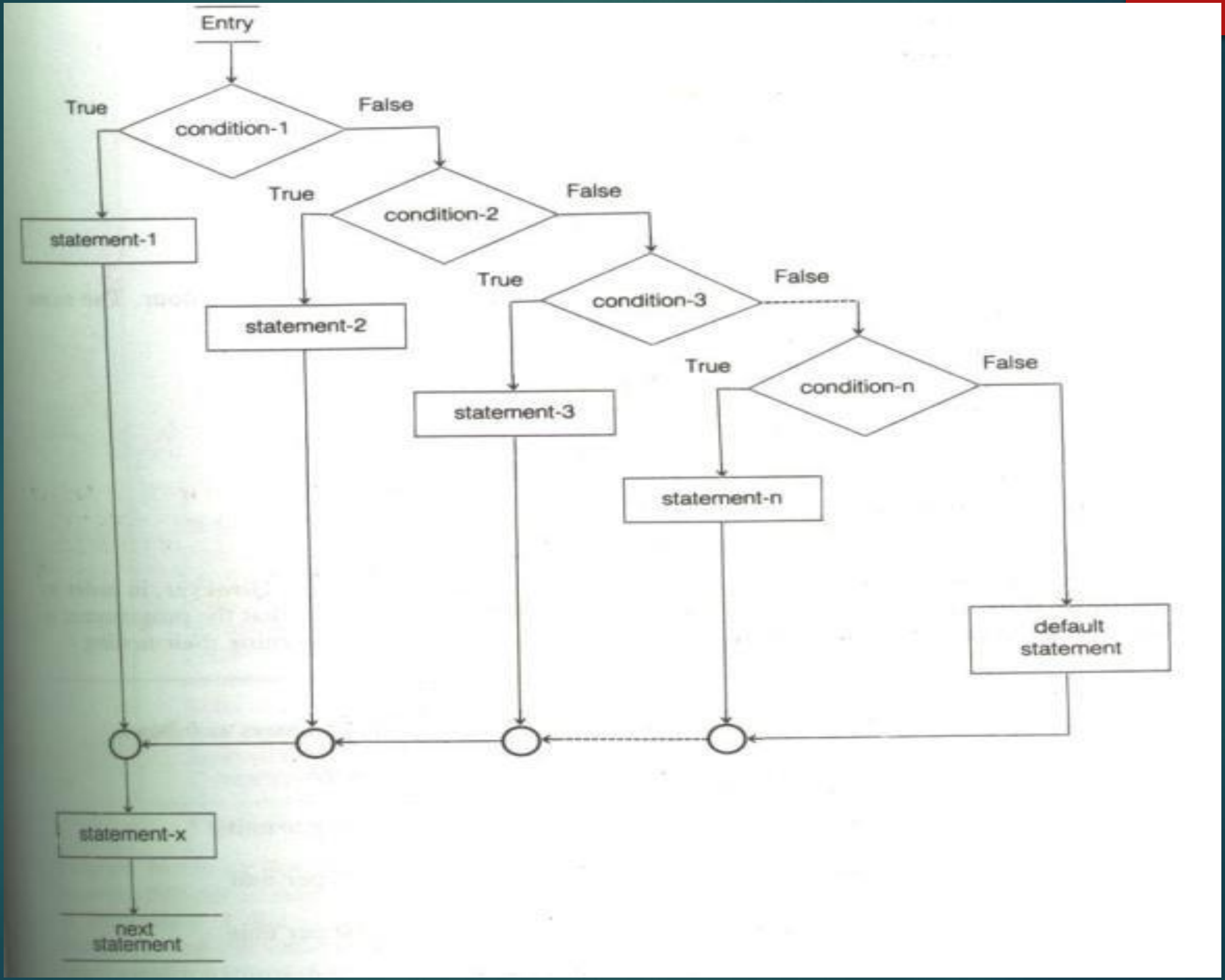
- A multiway decision is a chain of if conditions in which the statement associated with an else condition behaves like another if condition.

- Else if ladder is also called 3 way or multiway decision making statement.

The else if ladder(Syntax)

```
if (condition1)
{
   //These statements would execute if the condition1 is true
}
else if(condition2)
{
   //These statements would execute if the condition2 is true
}
else if (condition3)
{
   //These statements would execute if the condition3 is true
}
.
.
else
{
   //These statements would execute if all the conditions return false.
}
```

FLOWCHART OF ELSE....IF LADDER



```
// C program to illustrate else-if ladder

#include <stdio.h>

int main()
{
    int i = 20;

    // Check if i is 10
    if (i == 10)
        printf("i is 10");

    // Since i is not 10
    // Check if i is 15
    else if (i == 15)
        printf("i is 15");

    // Since i is not 15
    // Check if i is 20
    else if (i == 20)
        printf("i is 20");

// If none of the above conditions is true
    // Then execute the else statement
    else
        printf("i is not present");

    return 0;
}
```

```
Output:
i is 20
```

For example:

```
#include<stdio.h>
#include<conio.h>
void main( )
{
 int a;
 printf("enter a number");
 scanf("%d",&a);
 if( a%5==0 && a%8==0)
 {
 printf("divisible by both 5 and 8");
 }
 else if( a%8==0 )
 {
 printf("divisible by 8");
 }
 else if(a%5==0)
 {
 printf("divisible by 5");
 }
 else
 {
 printf("divisible by none");
 }
getch();
}
```

```
Output: enter a number
40
divisible by both 5 and 8
```