Conditional Statements

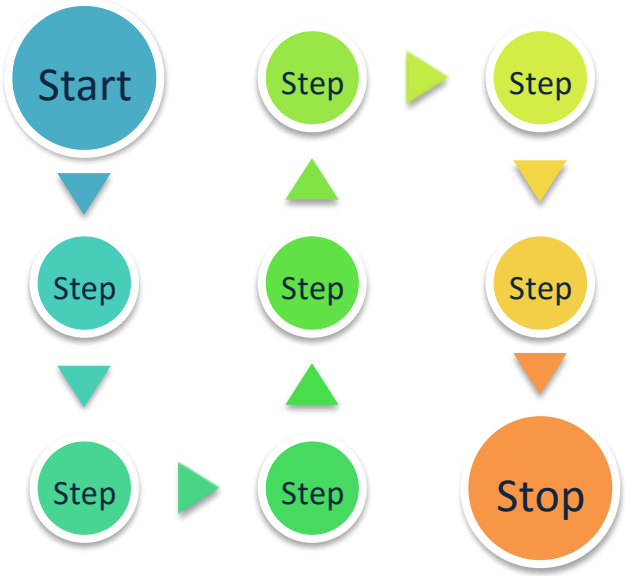
SESSION 5

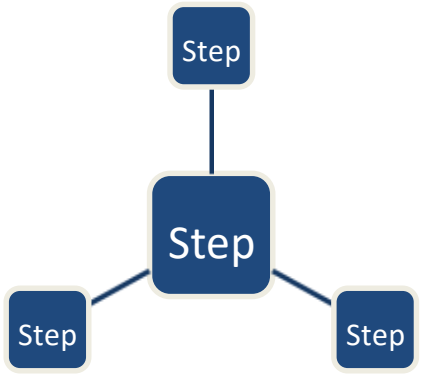
Objectives

- Explain IF statement
- Explain IF…ELSE selection construct
- Explain multiple selection statements
- Explain nested IF…ELSE statements
- Explain case construct

Introduction

- A programmer may come across a condition in the program, where the path of execution can branch into two or more options.
- Such constructs are referred to as programming, selection, conditional, or branching constructs.





IF Statement 1-7

- The IF construct is a basic selection construct.
  - Consider an example where the customer is given a discount if purchases of over $100 are made.
  - Each time a customer is billed, a part of the code has to check to see if the bill amount exceeds $100.
  - If it does exceed the amount, then it must deduct 10% of the total amount, otherwise nothing must be deducted.

IF Statement 2-7

- The pseudocode for the scenario will be as follows:

```
IF customer purchases items worth more than $100
  Give 10% discount
```

IF Statement 3-7

- The general form of an IF statement or construct is as follows:

```
IF condition
    Statements } Body of the IF Construct
END IF
```

IF Statement 4-7

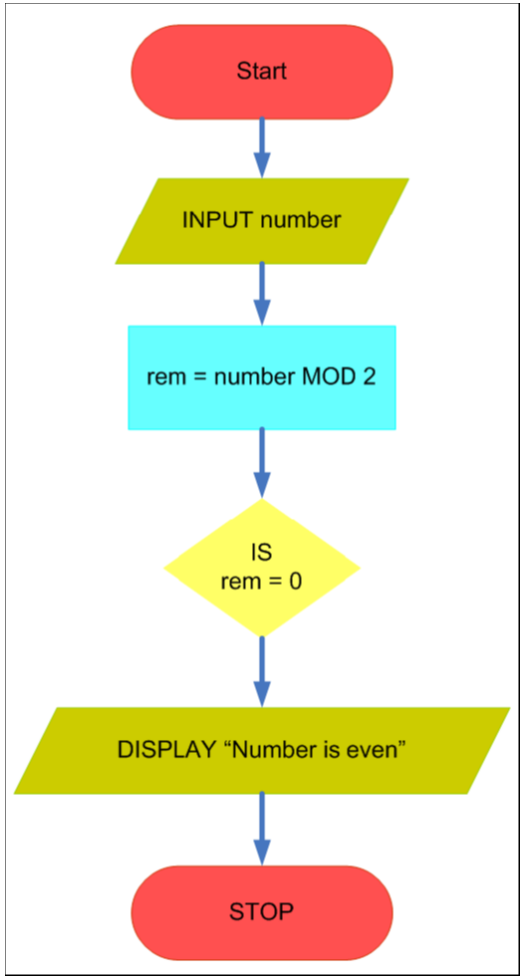
- The example uses the IF construct to find whether a number is even or not.

```
BEGIN
  INPUT number
  rem = number MOD 2
  IF rem=0
   Display "Number is even"

  END IF
END
```

IF Statement 5-7

- A flowchart for the pseudocode is shown in the figure.



IF Statement 6-7

- The syntax for the IF statement in C# language is as follows:

```
if (condition)
  {
  Statements;
  }
```

IF Statement 7-7

- The example shows the pseudocode that would be written in C#

```
public static void main (String[] args)
{
int number, rem;
Console.WriteLine("Please enter a number: ");
Number = Convert.ToInt32(Console.ReadLine());


rem=number%2;
if(rem==0)
   {
   Console.WriteLine("Even Number");
   }
}
```

IF…ELSE Statement 1-4

- The IF…ELSE statement enables a programmer to make a single comparison, and then execute the steps depending on whether the result of the comparison is true or false.
- The general form of the IF…ELSE statement is as follows:

```
IF condition
  Statement set1
ELSE
  Statement set2
END IF
```

IF…ELSE Statement 2-4

- The syntax for the IF…ELSE construct in C# language is given as follows:

```
if(condition)
   {
   statement set1;
   }
else
   {
   statement set2;
   }
```

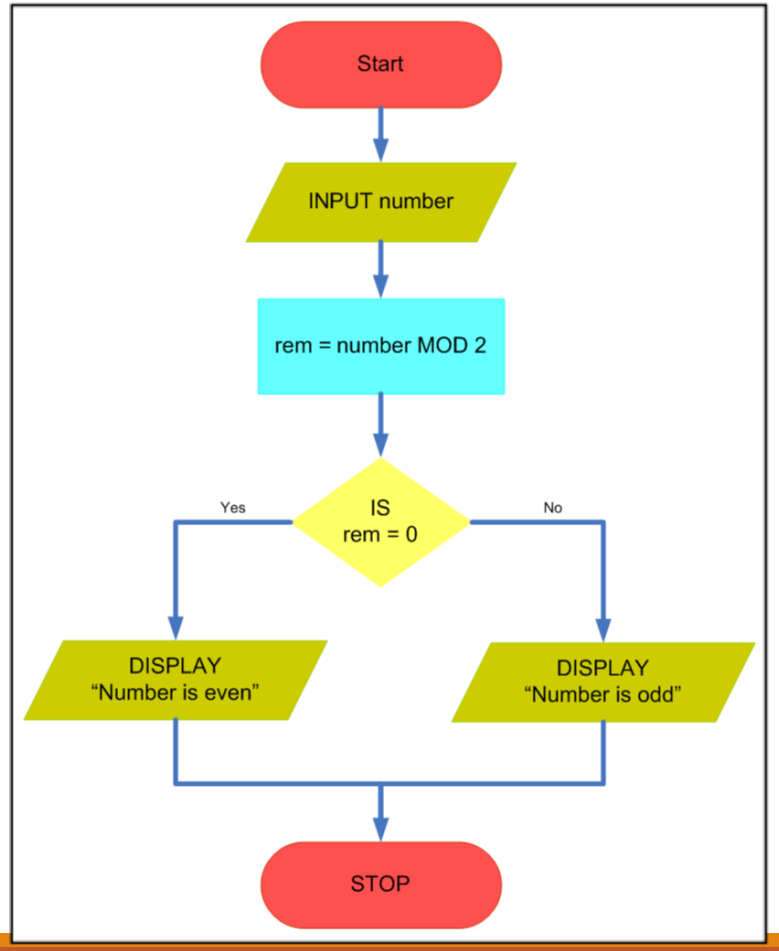
IF…ELSE Statement 3-4

- A more efficient code for the even number using the IF…ELSE statement is shown in the following example.

```
BEGIN
INPUT number
rem=number MOD 2
IF rem=0
  DISPLAY "Even Number"
ELSE
  DISPLAY "Odd Number"
END IF
END
```

IF...ELSE Statement 4-4

- The flowchart for the pseudocode is shown:



Multiple Selection Statements 1-3

- The AND statement can be used in conjunction with the IF statement for more than one condition.
- To classify a supplier as a Most Valuable Supplier (MVS), the organization must check that the supplier has been with them for the last 10 years.
- And has done a total business of more than $500000.
- These two conditions must be satisfied to consider a supplier as a MVS.

Multiple Selection Statements 2-3

- The example shows the pseudocode for this scenario.

```
BEGIN
INPUT YearsWithUs
INPUT BizDone
IF YearsWithUs >= 10 AND BizDone >= 500000
  DISPLAY "Classified as an MVS"
ELSE
  DISPLAY "A little more effort required"
END IF
END
```

Multiple Selection Statements 3-3

- The example shows the pseudocode that would be written in C#.

```
/* C# snippet depicting the AND operator in IF */
if(YearsWithUs >= 10 && BizDone >= 500000)
   {
   Console.WriteLine("Classified as an MVS");
   }
else
   {
   Console.WriteLine("A little more effort required");
   }
```

Nested IF…ELSE Statements 1-3

- Another way to combine two conditions without using the AND operator, is by using nested IF…ELSE statements.

- A nested IF is an IF statement written inside another IF statement.

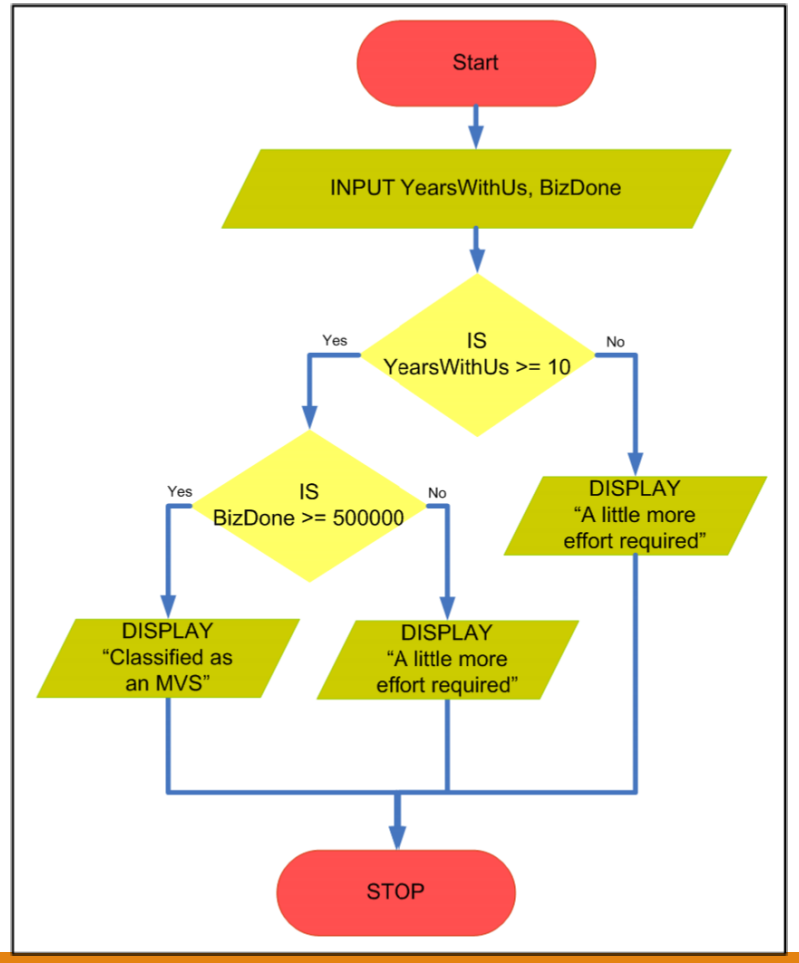
Nested IF…ELSE Statements 2-3

- Consider the earlier example to recognize the MVS status of a supplier rewritten using nested IF.

```
BEGIN
INPUT YearsWithUS
INPUT BizDone
IF YearsWithUs >= 10
   IF BizDone >= 500000
      DISPLAY "Classified as an MVS"
   ELSE
      DISPLAY "A little more effort required"
   END IF
ELSE
   DISPLAY "A little more effort required"
END IF
END
```

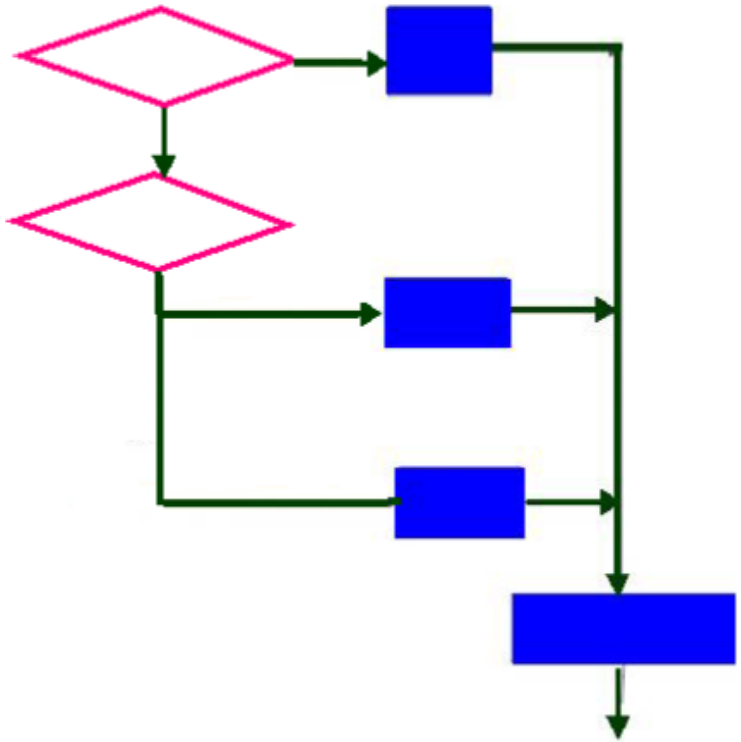
Nested IF...ELSE Statements 3-3

- The flowchart for the pseudocode is shown in the figure.



Case Conditions 1-2

- The DO CASE…END CASE construct is used when a variable is to be successively compared against different values.
- The DO CASE is known as ‘Switch Case’ in C#.



Case Conditions 2-2

- The syntax in C will be as follows:

```
switch (expression)
{
  case const-expr:
    statement set;
    break;
  case const-expr:
    statement set;
    break;
  default
    statement set;
}
```

A Switch Statement

```
switch (ch) {
 case 'a': case 'A':
 case 'e': case 'E':
 case 'i': case 'I':
 case 'o': case 'O':
 case 'u': case 'U':
       Console.WriteLine(ch + " is a vowel" );
       break;
 default:
       Console.WriteLine(ch + " is not a vowel" );
}
```