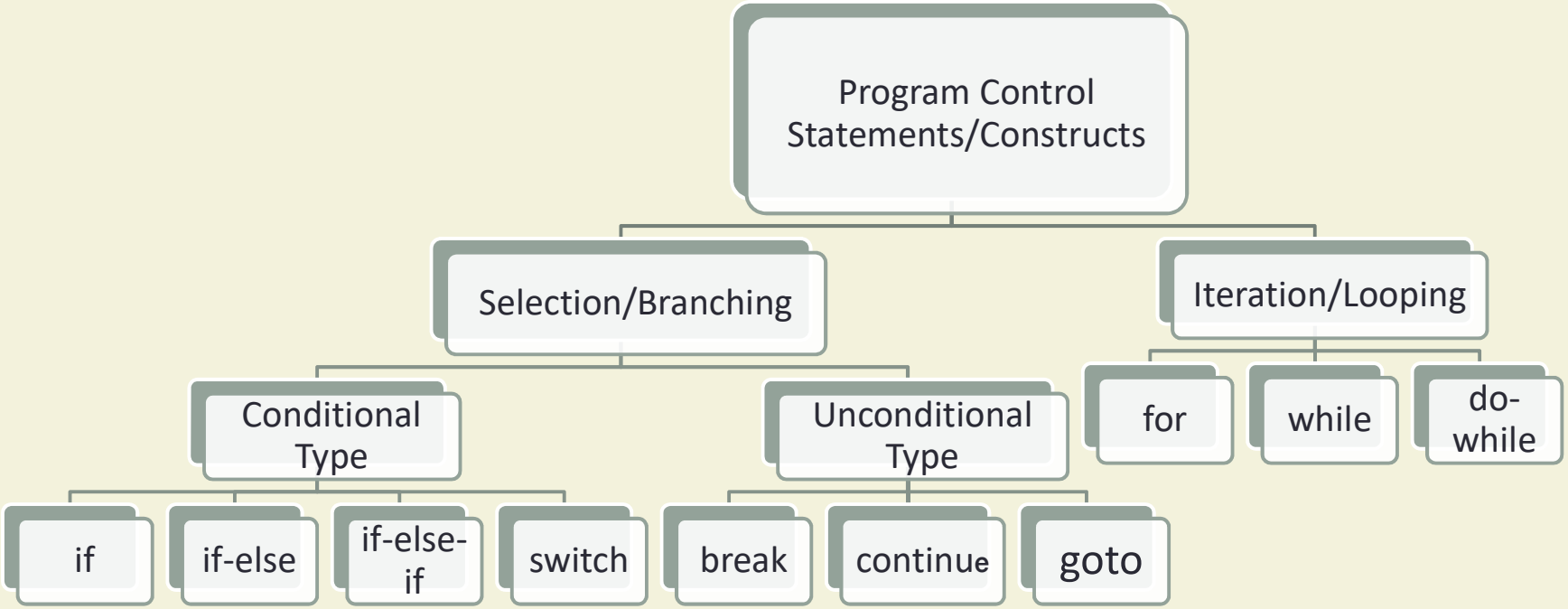
CONTROL STATEMENTS

L.DIVAKARARAO
Assistant Professor
Department of Computer Science
Aditya Degree College
Kakinada

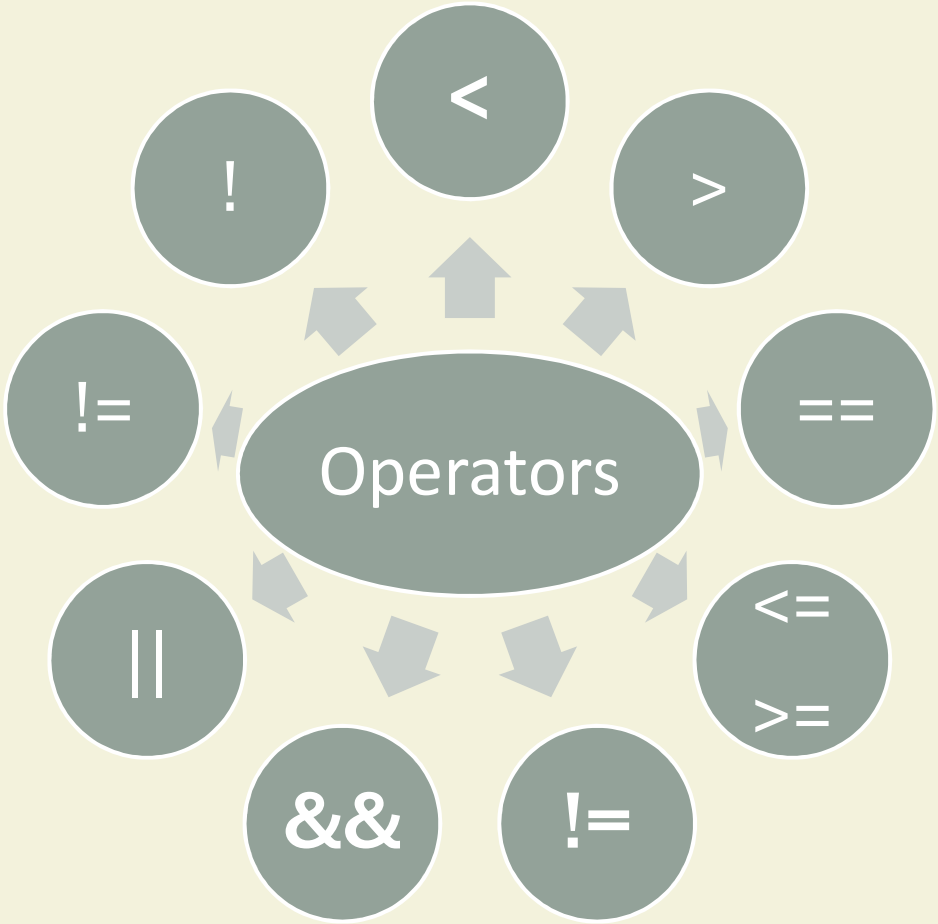
CONTROL STATEMENTS INCLUDE

Selection Statements	Iteration Statements	Jump Statements
• if	• for	• goto
• if-else	• while	• break
• switch	• do-while	• continue
		• return

CONTROL STATEMENTS



OPERATORS



A FEW EXAMPLES

The following declarations and initializations are given:

```
int x=1, y=2, z=3;
```

Then,

- The expression x>=y evaluates to 0 (false).
- The expression x+y evaluates to 3 (true).
- The expression x=y evaluates to 2 (true).

CONDITIONAL EXECUTION AND SELECTION

- Selection Statements
- The Conditional Operator
- The switch Statement

SELECTION STATEMENTS

One-way decisions using if statement

Two-way decisions using if-else statement

Multi-way decisions

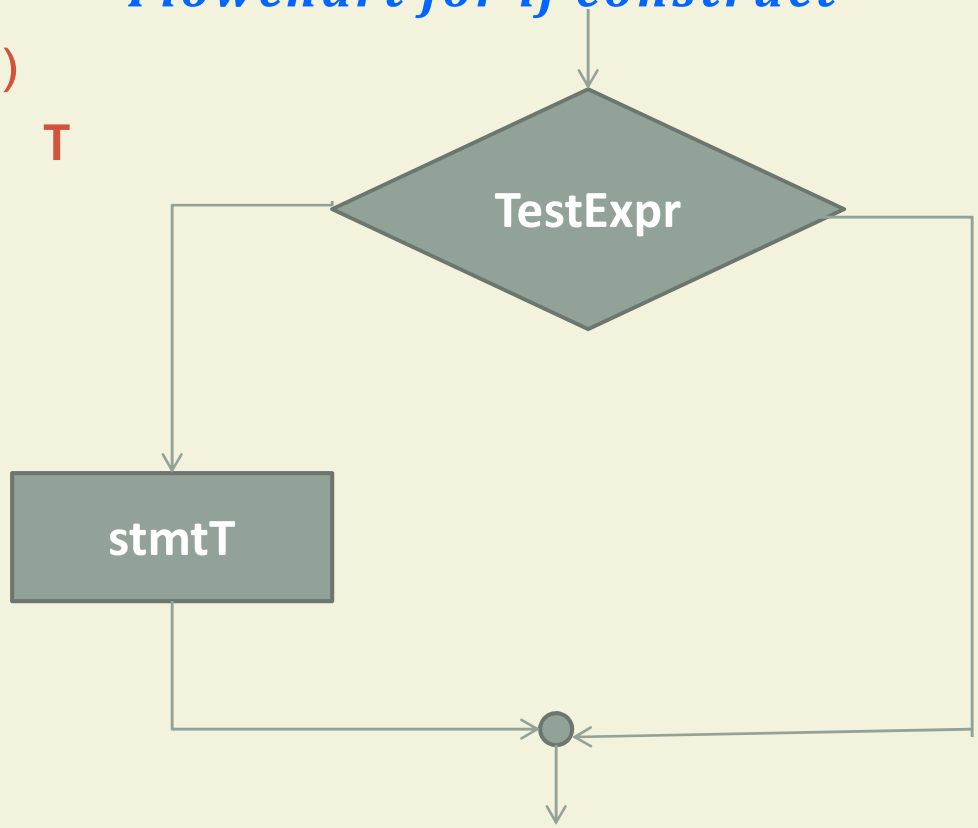
Dangling else Problem

ONE-WAY DECISIONS USING IF STATEMENT

Flowchart for if construct

```
if(TestExpr)
stmtT;
```

T



WRITE A PROGRAM THAT PRINTS THE LARGEST AMONG THREE NUMBERS.

Algorithm	C Program

1. START	#include <stdio.h>
	int main()
	{
	int a, b, c, max;
	printf("\nEnter 3 numbers");
	scanf("%d %d %d", &a, &b, &c);
	max=a;
	if(b>max)
	max=b;
	if(c>max)
	max=c;
	printf("Largest No is %d", max);
	return 0;
	}

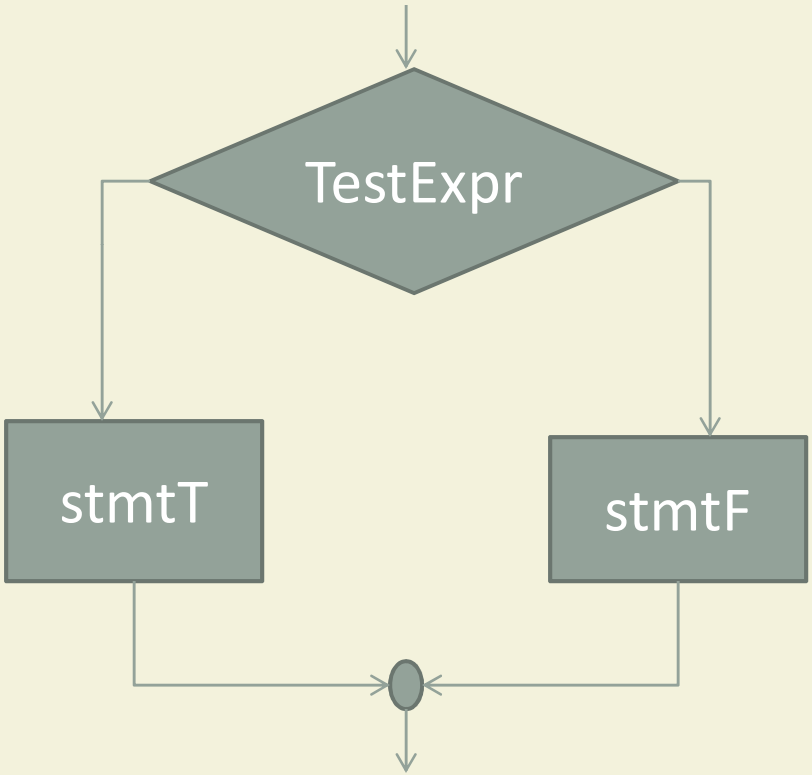
2. PRINT "ENTER THREE NUMBERS"	
3. INPUT A, B, C	
4. MAX=A	
5. IF B>MAX THEN MAX=B	
6. IF C>MAX THEN MAX=C	
7. PRINT "LARGEST NUMBER IS", MAX	
8. STOP	

TWO-WAY DECISIONS USING IF-ELSE STATEMENT

The form of a two-way decision is as follows:

```
if(TestExpr)
   stmtT;
else
   stmtF;
```

Flowchart of if-else construct



WRITE A PROGRAM THAT PRINTS THE LARGEST AMONG THREE NUMBERS.

Algorithm	C Program

1. START	#include <stdio.h>
	int main()
	{
	int a, b, c, max;
	printf("\nEnter 3 numbers");
	scanf("%d %d %d", &a, &b, &c);
	max=a;
	if(b>max)
	max=b;
	if(c>max)
	max=c;
	printf("Largest No is %d", max);
	return 0;
	}

2. PRINT "ENTER THREE NUMBERS"	
3. INPUT A, B, C	
4. MAX=A	
5. IF B>MAX THEN MAX=B	
6. IF C>MAX THEN MAX=C	
7. PRINT "LARGEST NUMBER IS", MAX	
8. STOP	

MULTI-WAY DECISIONS

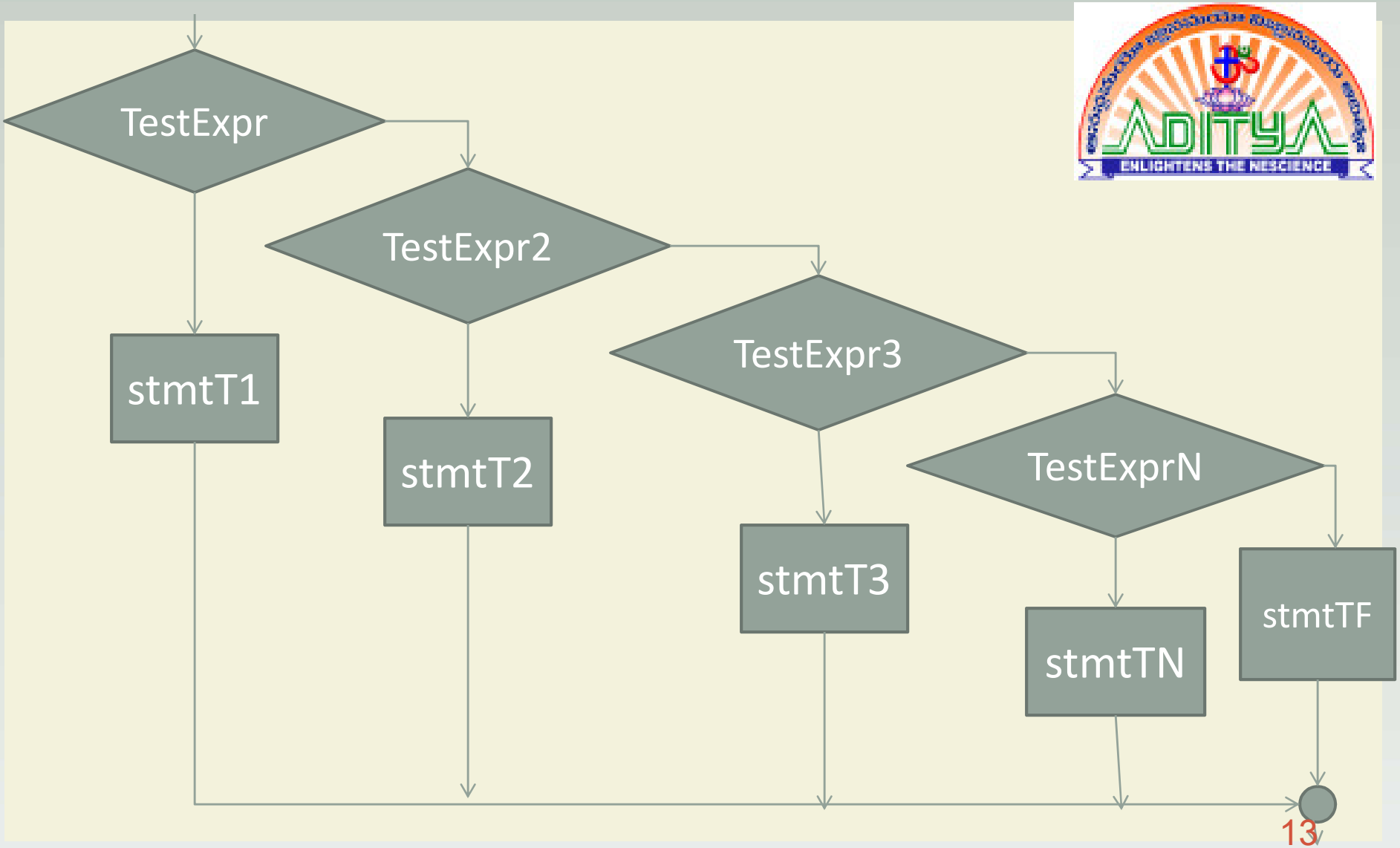
```
if(TestExpr1)
  stmtT1;
    else if(TestExpr2)
      stmtT2;
        else if(TestExpr3)
          stmtT3;
            . . .
              else if(TestExprN)
                stmtTN;
                  else
                    stmtF;
```

if-else-if ladder

```
switch(expr)
{
  case constant1: stmtList1;
    break;
  case constant2: stmtList2;
    break;
  case constant3: stmtList3;
    break;
    ……————————————————.
    ……————————————————.
  default: stmtListn;
}
```

General format of switch statements

FLOWCHART OF AN IF-ELSE-IF CONSTRUCT



THE FOLLOWING PROGRAM CHECKS WHETHER A NUMBER GIVEN BY THE USER IS ZERO, POSITIVE, OR NEGATIVE

```
#include <stdio.h>
int main()
{
  int x;
  printf("\n ENTER THE NUMBER:");
  scanf("%d", &x);
  if(x > 0)
     printf("x is positive \n");
       else if(x == 0)
         printf("x is zero \n");
           else
            printf("x is negative \n");
             return 0;
}
```

NESTED IF

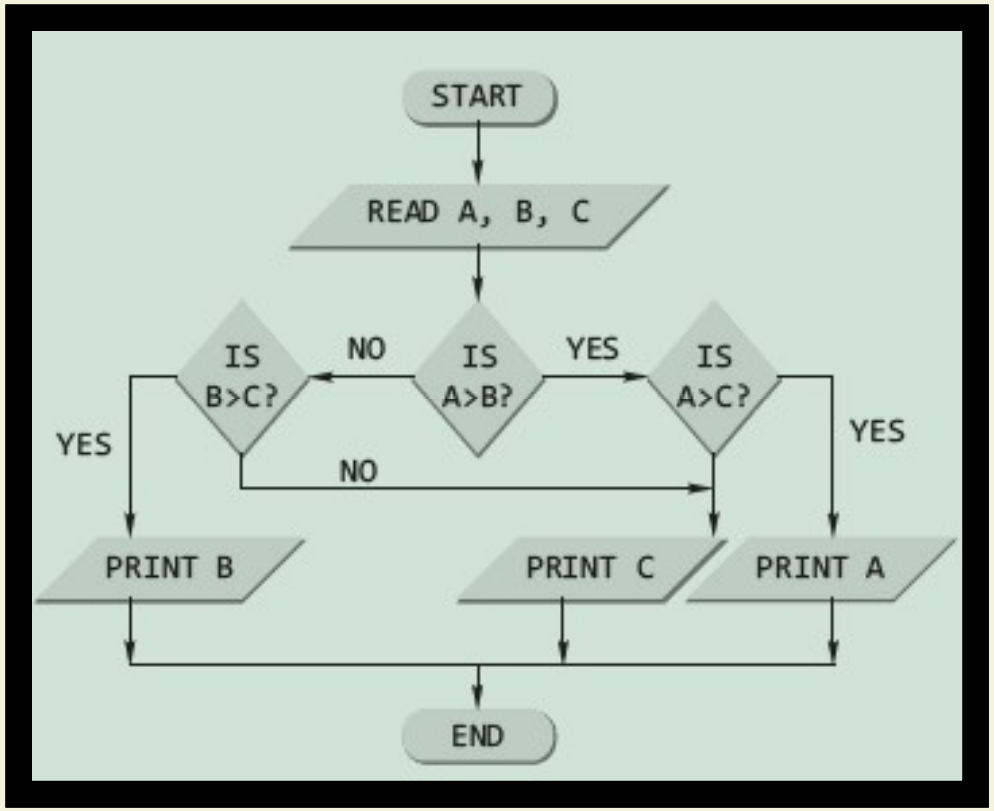
- When any if statement is written under another if statement, this cluster is called a nested if.
- The syntax for the nested is given here:

Construct 1	Construct 2

if(TestExprA)
  if(TestExprB)
  stmtBT;
  else
  stmtBF;
else
  stmtAF;
	if(TestExprA)
	  if(TestExprB)
	  stmtBT;
	  else
	  stmtBF;
	else
	  if(TestExprC)
	  stmtCT;
	  else
	  stmtCF;

A PROGRAM TO FIND THE LARGEST AMONG THREE NUMBERS USING THE NESTED LOOP

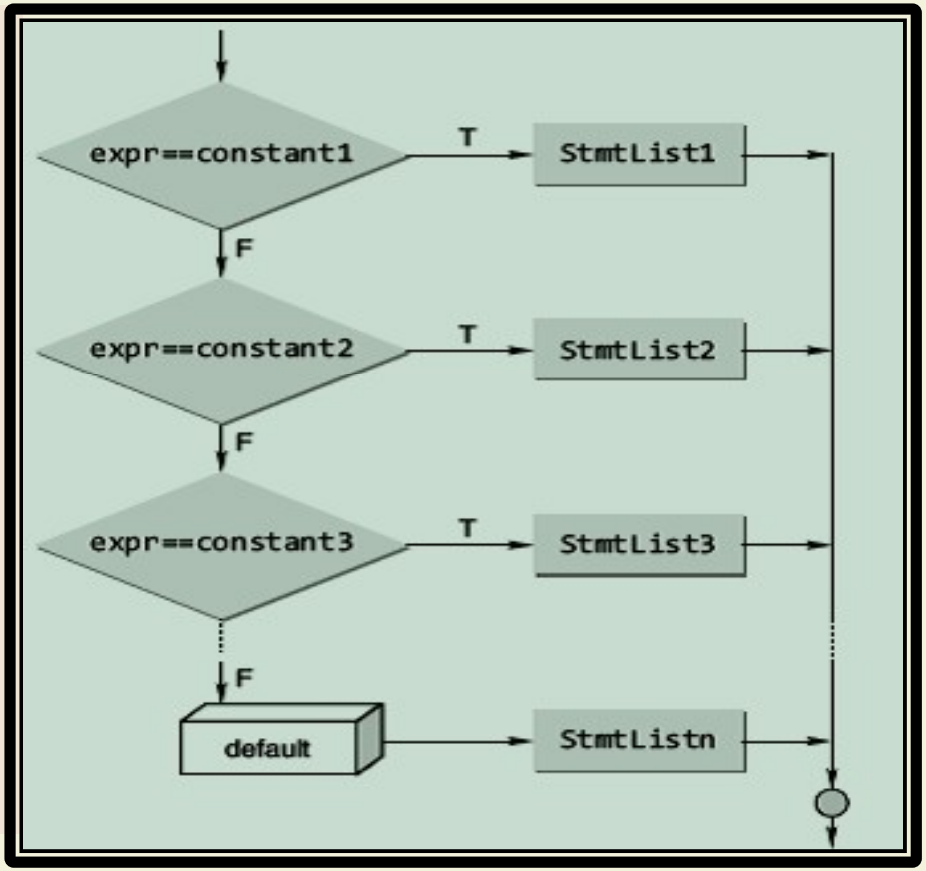
```
#include <stdio.h>
int main()
{
  int a, b, c;
  printf("\nEnter the three numbers");
  scanf("%d %d %d", &a, &b, &c);
  if(a > b)
     if(a > c)
        printf("%d", a);
     else
         printf("%d", c);
  else
     if(b > c)
         printf("%d", b);
     else
         printf("%d", c);
  return 0;
}
```



THE SWITCH STATEMENT

The general format of a switch statement is

```
switch(expr)
{
case constant1: stmtList1;
break;
case constant2: stmtList2;
break;
case constant3: stmtList3;
break;
……————————————————.
……————————————————.
default: stmtListn;
}
```



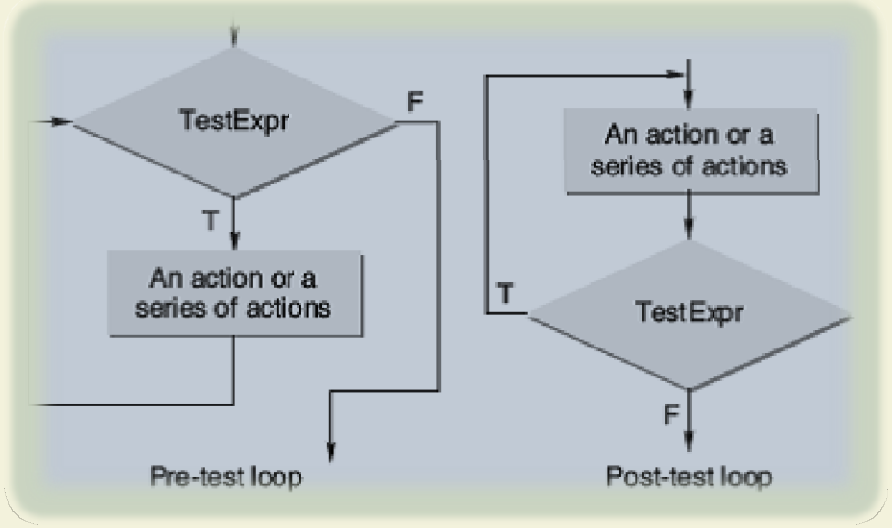
The C switch construct

SWITCH VS NESTED IF

- The switch differs from the else-if in that switch can test only for equality, whereas the if conditional expression can be of a test expression involving any type of relational operators and/or logical operators.
- A switch statement is usually more efficient than nested ifs.
- The switch statement can always be replaced with a series of else-if statements.

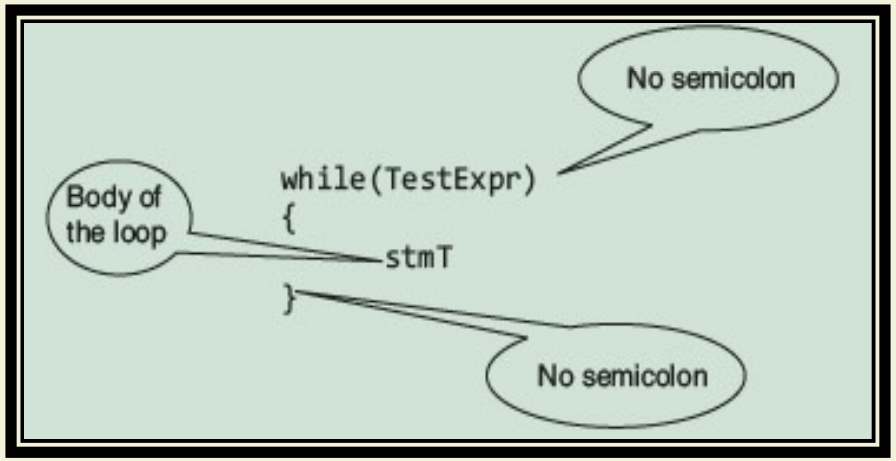
ITERATION AND REPETITIVE EXECUTION

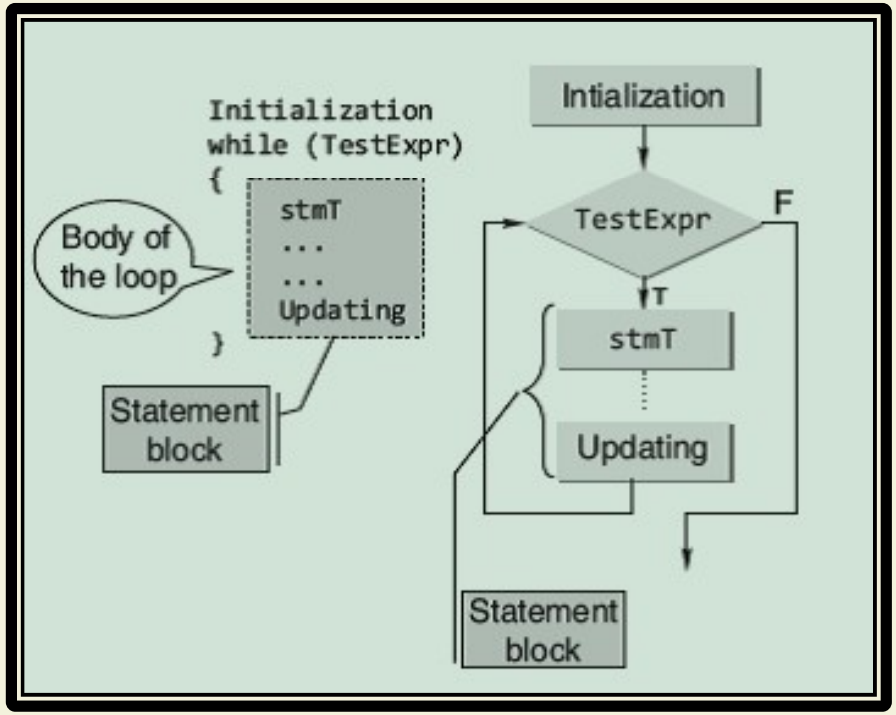
- A loop allows one to execute a statement or block of statements repeatedly. There are mainly two types of iterations or loops – *unbounded iteration or unbounded loop and bounded iteration or bounded loop.*
- A loop can either be a *pre-test loop or be a post-test loop* as illustrated in the diagram.



"WHILE" CONSTRUCT

while statement is a pretest loop. The basic syntax of the while statement is shown below:





AN EXAMPLE

```
#include <stdio.h>
int main()
{
  int c;
  c=5; // Initialization
  while(c>0)
     { // Test Expression
       printf(" \n %d",c);
       c=c-1; // Updating
     }
   return 0;
}
```

This loop contains all the parts of a while loop. When executed in a program, this loop will output

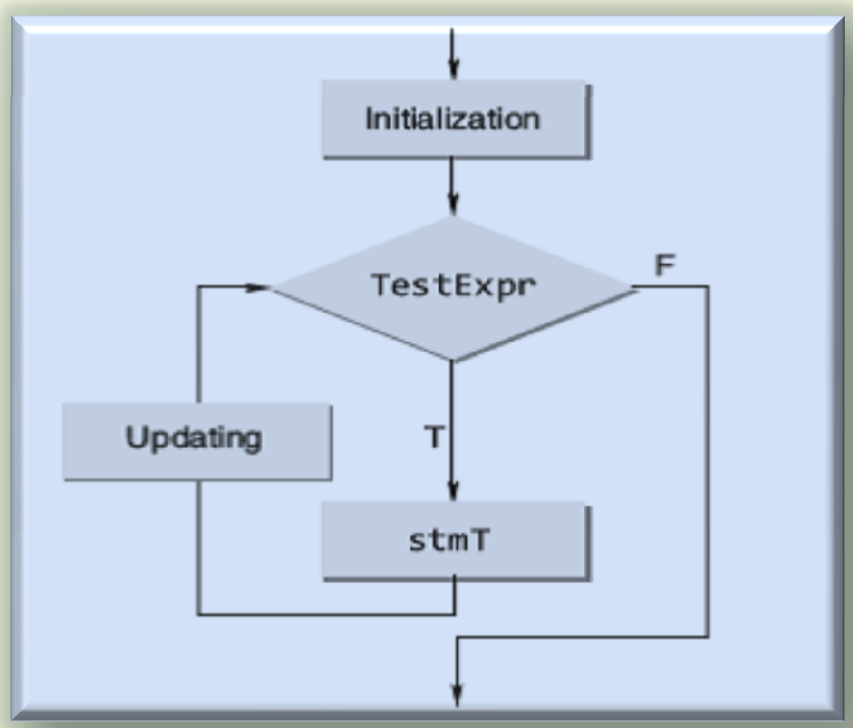
5
4
3
2
1

"FOR" CONSTRUCT

- The general form of the for statement is as follows:

```
for(initialization; TestExpr; updating)
   stmT;
```

for construct flow chart



EXAMPLE

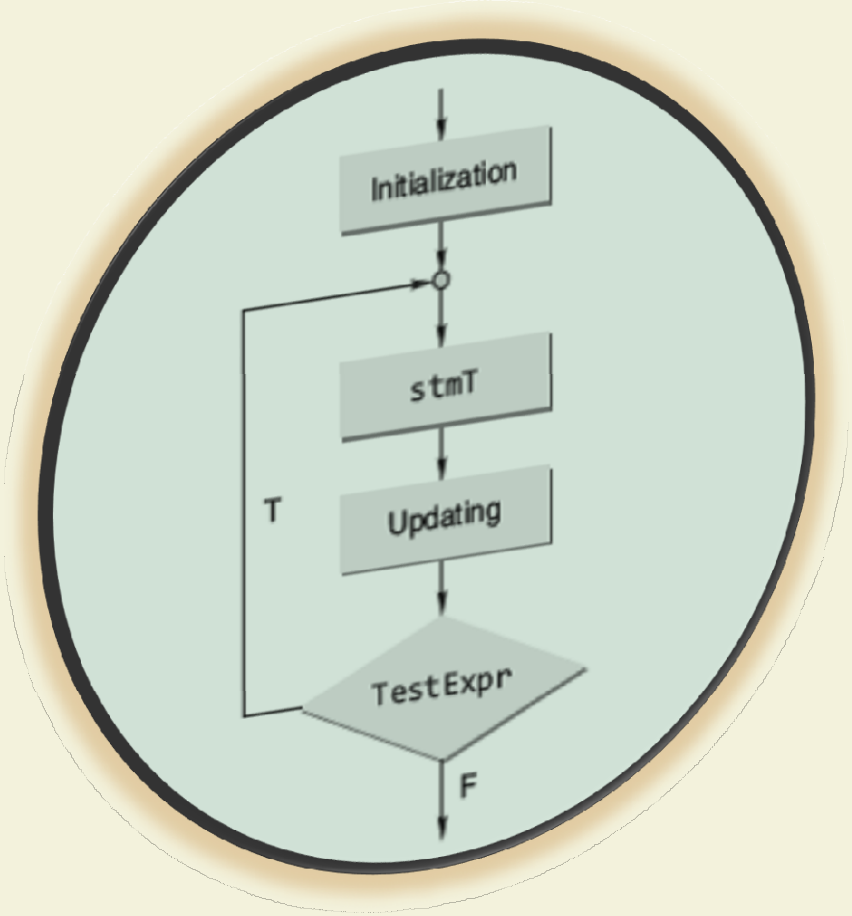
```
#include <stdio.h>
int main()
{
   int n, s=0, r;
   printf("\n Enter the Number");
   scanf("%d", &n);
   for(;n>0;n/=10)
      {
        r=n%10;
        s=s+r;
      }
    printf("\n Sum of digits %d", s);
    return 0;
}
```

"DO-WHILE" CONSTRUCT

The C do-while loop

The form of this loop construct is as follows:

```
do
{
  stmT; /* body of statements would be placed here*/
}while(TestExpr);
```



AN EXAMPLE

```
#include <stdio.h>
int main()
{
  int x = 1;
  int count = 0;
  do {
        scanf("%d", &x);
        if(x >= 0) count += 1;
      } while(x >= 0);
  return 0;
}
```

GOTO STATEMENT

The control is unconditionally transferred to the statement associated with the label specified in the goto statement. The form of a goto statement is

goto label_name;

The following program is used to find the factorial of a number.

```
#include <stdio.h>
int main()
{
  int n, c;
  long int f=1;
  printf("\n Enter the
  number:");
  scanf("%d",&n);
  if(n<0)
     goto end;
  for(c=1; c<=n; c++)
      f*=c;
  printf("\n FACTORIAL IS %ld",
  f);
  end:
   return 0;
 }
```

SPECIAL CONTROL STATEMENTS

- "return" statements
- "break" statements
- "continue" statements