If Control Construct

A mechanism for deciding whether an action should be taken

Conditional Constructs

- Provide
  - Ability to control whether a statement list is executed
- Two constructs
  - If statement
    - if
    - if-else
    - if-else-if
  - Switch statement

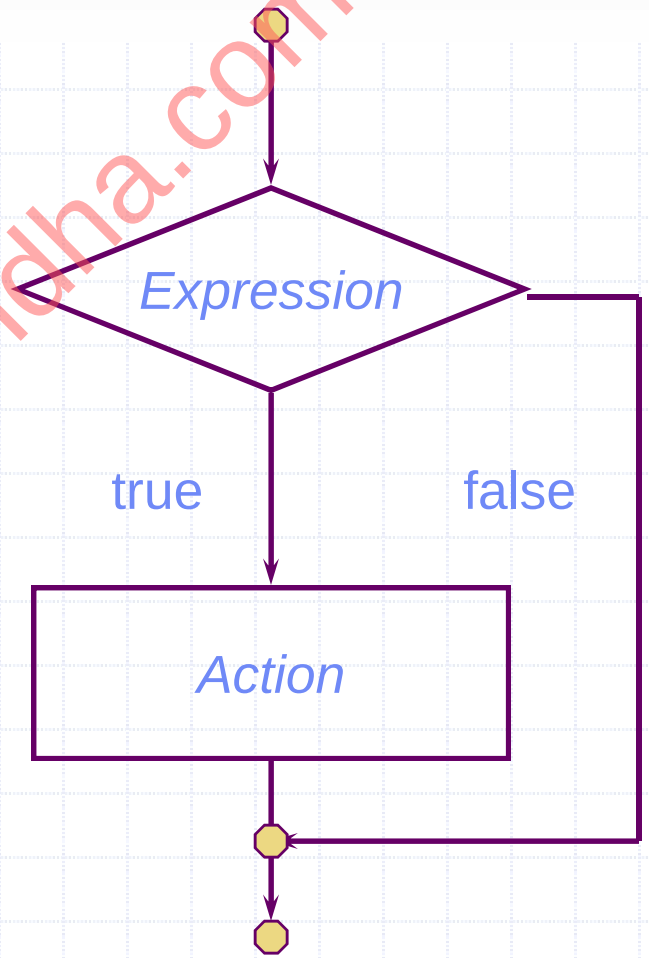
The Basic If Statement

- Syntax

  ```
  if (Expression)
      Action
  ```

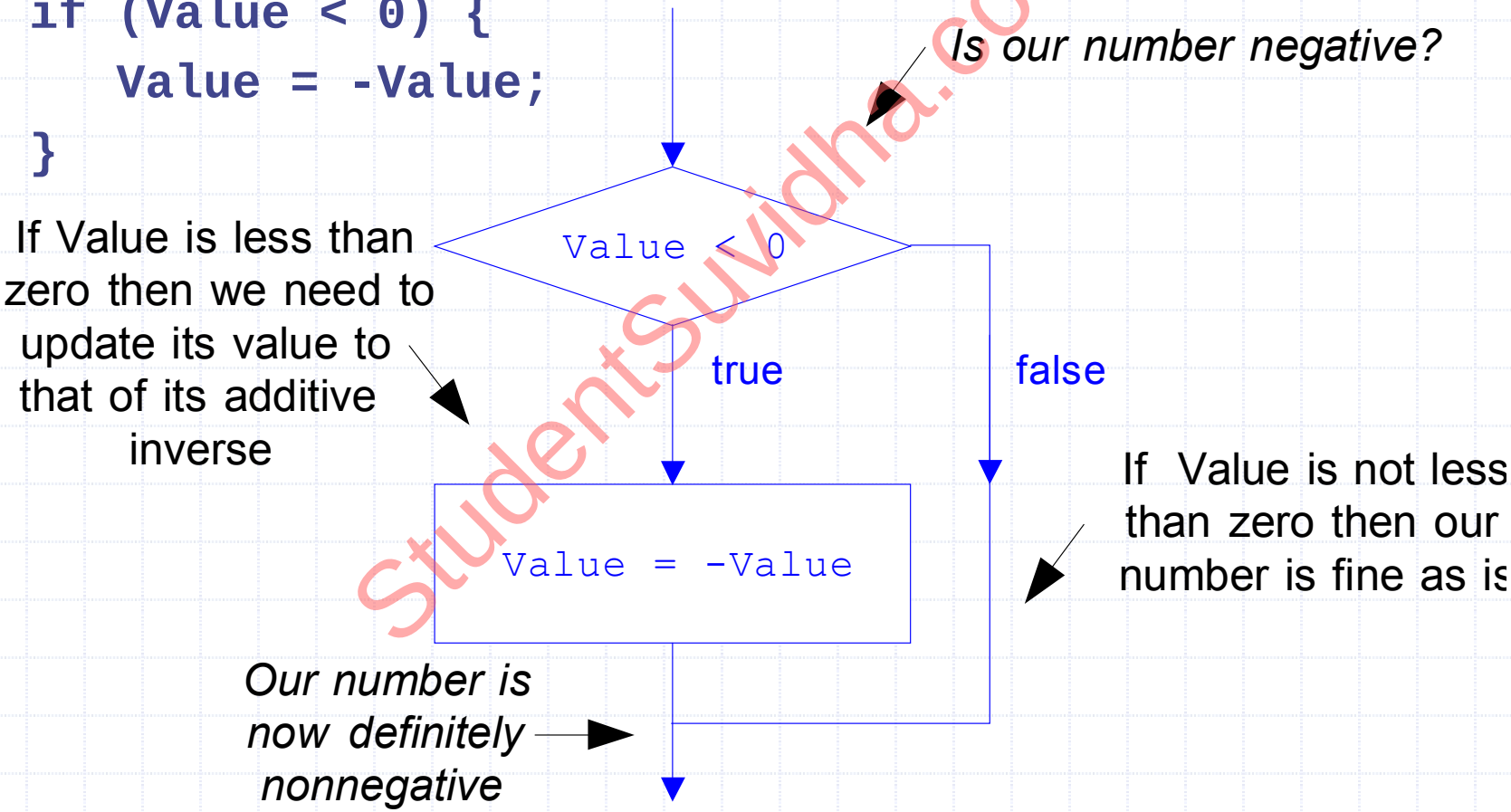
- If the Expression is true then execute Action

- Action is either a single statement or a group of statements within braces



Example

```
if (Value < 0) {
    Value = -Value;
}
```



Example of If

```
#include<stdio.h>
 void main()
 {
  int mynumber;
  scanf("%d",&mynumber);
  if(mynumber==10)
  printf("Isequal\n")
}
```

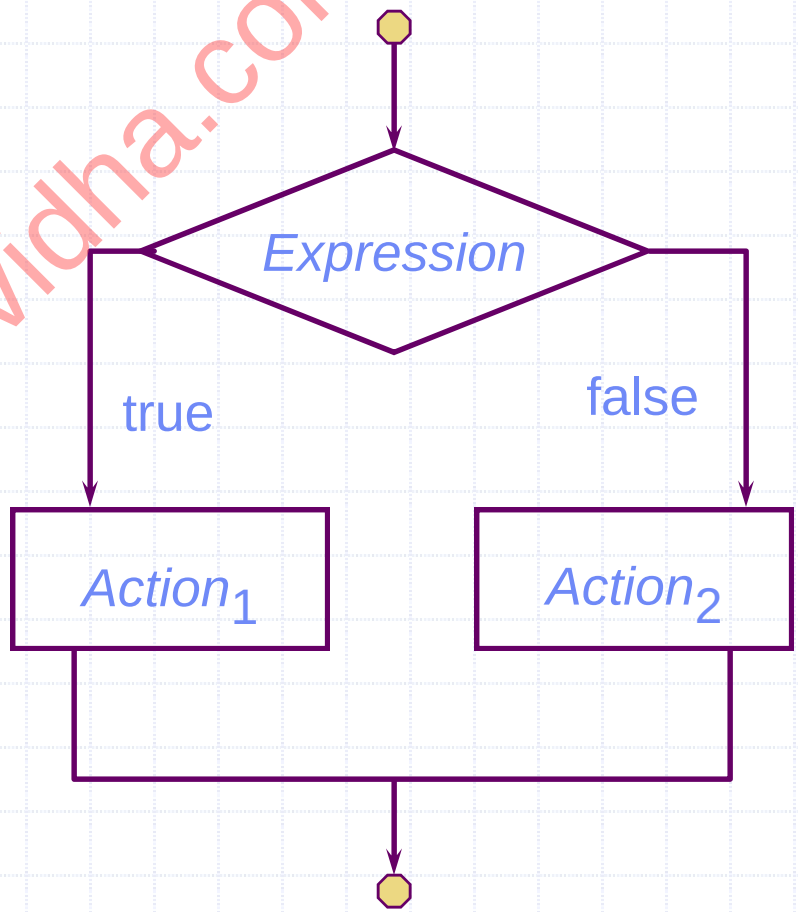
The If-Else Statement

- Syntax

```
if (Expression)
        Action1
    else
        Action2
```

- If Expression is true then execute $Action_1$ otherwise execute $Action_2$

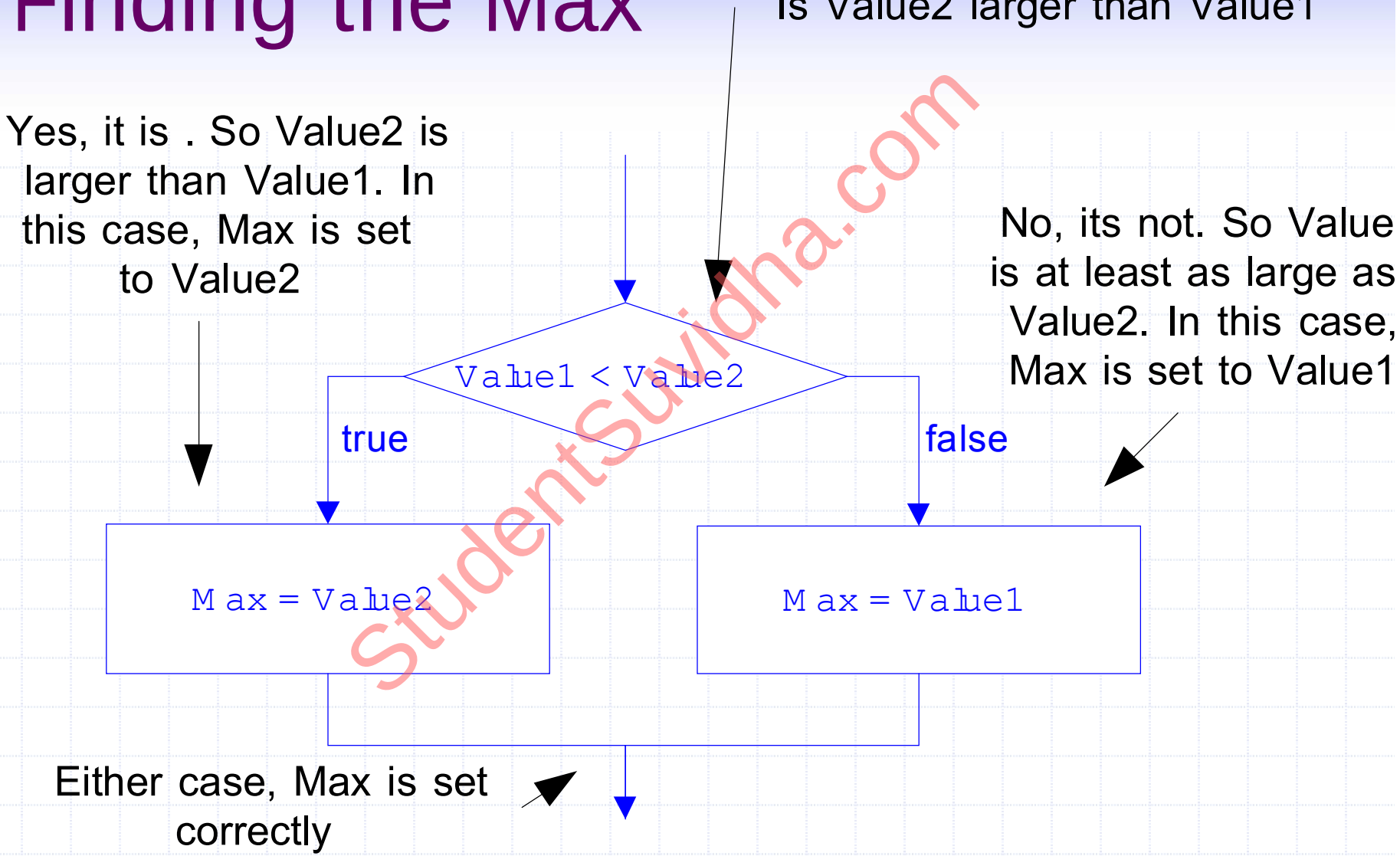
```
if (v == 0) {
    cout << "v is 0";
  }
  else {
    cout << "v is not 0";
  }
```



Finding the Max

```
#include<stdio.h>
#include<conio.h>
void main()
{
Printf(“Enter two integers: “);
int Value1;
int Value2;
Scanf(“%d%d”, &Value1, &Value2);
int Max;
if (Value1 < Value2) {
    Max = Value2;
}
else {
    Max = Value1;
}
Printf("Maximum of inputs is: %d“,  Max);
getch();
}
```

Finding the Max



Selection

- It is often the case that depending upon the value of an expression we want to perform a particular action
- Two major ways of accomplishing this choice
  - if-else-if statement
    - if-else statements “glued” together
  - Switch statement
    - An advanced construct

An If-Else-If Statement

```
if ( nbr < 0 )
{
   printf("%d is negative", nbr);
}
else if ( nbr > 0 )
{
   printf("%d is positive", nbr);
}
else
{
   printf("%d is zero", nbr);
}
```

A Switch Statement

- A multiple selection structure is useful when an algorithm contains a series of decisions in which a variable or expression is tested separately for one of several possible integral values.
- Each integral value represents a different action to be taken in the algorithm.
- C provides the switch multiple selection structure to implement this type of decision making.

Switch-Case Structures

- The switch - case syntax is:

```
switch (integer expression test value)
{
    case  case _1_fixed_value :
        action(s) ;
    case  case_2_fixed_value :
        action(s) ;
   default :
        action(s) ;
}
```

Switch-Case Structures

- The switch is the "controlling expression"
  - Can only be used with constant integer expressions.
  - Remember, a single character is a small positive integer.
  - The expression appears in ( )
- The case is a "label“
  - The label must be followed by a " : "
  - Braces, { }, not required around statements

A Switch Statement

```
void main()
{ char ch;
Printf(“Enter any character”);
Scanf(“%c”,&ch);
switch (ch) {
   case 'a': case 'A':
   case 'e': case 'E':
   case 'i': case 'I':
   case 'o': case 'O':
   case ’u’: case ’U’: printf(“ %c ia a vowel”, ch);
                       break;
   default: printf(“%c is not a vowel“,ch);
}
}
```

Assignment

- Use a switch-case structure to select from among the shapes for which calculations are to be made.May use just first character of shape name to select which calculation to make.Program only does one shape, and then exits.