



DECISION MAKING STATEMENTS

If, else if, switch case

Decision Making Statements

- These statements allow the execution of *selective statements* based on certain **decision criteria**.
- C language **provides the following statements**:
 - **if statement**
 - **if - else statement**
 - Nesting (i.e., embedding) if else statements
 - **switch statement**

Decision Making Statements

- **if** statement
 - Till now, we have written programs in which **instructions were executed in order of their appearance.**
 - There will be **many situations** where such **execution sequence may not be sufficient.**
 - For example, we want to find the bigger out of two numbers. This problem requires comparison of two numbers and based on comparison result, **the bigger number will be found.**
 - This can be implemented using **if** instruction.

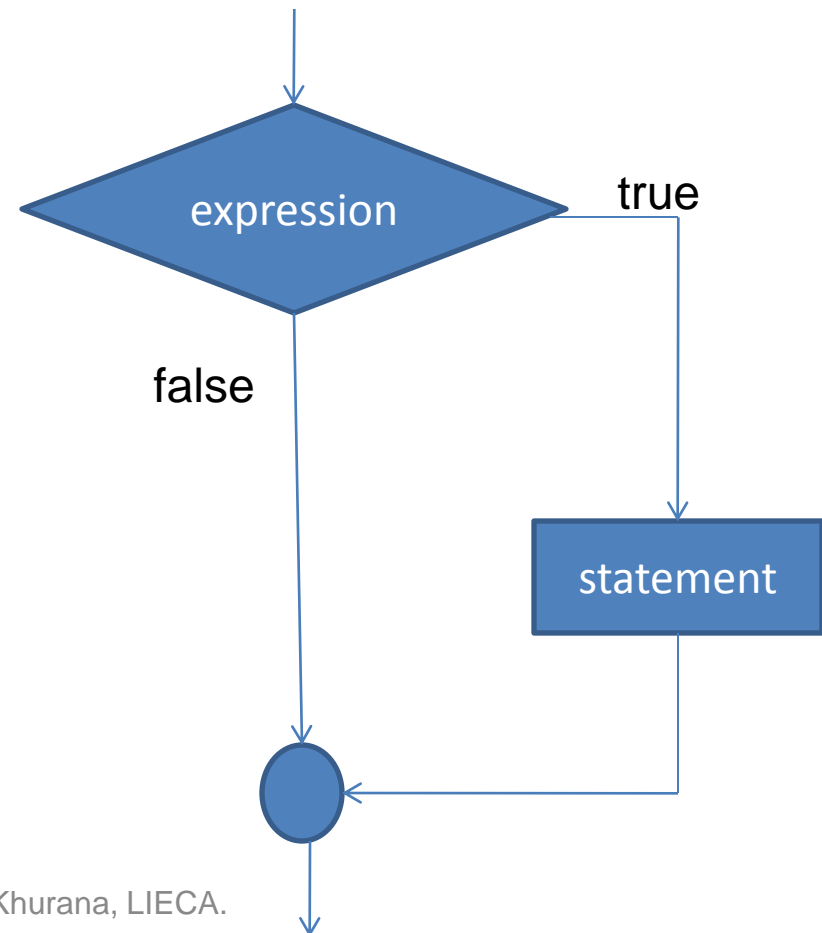
Decision Making Statements

- **if** statement
 - **if instruction** carries out a logical test and then takes one of the two possible actions depending on the outcome of the test (**i.e., whether the outcome is *true* or *false***).
 - This instruction can be written in simplest form as:
 - if (**expr**) statement
 - » The **expr** must be placed in **parenthesis**, as shown. In this form, the statement will be executed only if the **expr** has a **nonzero value** (i.e., if **expr** is **true**).
 - » If the **expr** has a value of **zero** (i.e., if the **expr** is **false**), then the statement will be ignored.

Decision Making Statements

- Code of *if* statement and Logic flow control

```
:  
:  
:  
If ( expression )  
{  
    statement  
}  
:  
:
```



Decision Making Statements

- if statement - Example

- Let us try to use *if instruction* to find out bigger out of two numbers.

```
#include<stdio.h>
void main(void)
{
int first, second;
scanf( " %d %d ", &first, &second);
if ( first > second )
    printf ( "First number is bigger\n");
}
```

- In the program, if the input value of first number is higher than second number, **only then the printf statement will be executed.**

Decision Making Statements

- if statement - Example

- Let us try to use *if instruction* to find out bigger out of two numbers.

```
#include<stdio.h>
void main(void)
{
int first, second;
scanf( " %d %d ", &first, &second);
if ( first > second )
    printf ( "First number is bigger\n");
}
```

- So if the inputs are given as:

20 10

- We will see the output
 - First number is bigger

Decision Making Statements

- **if** statement - Example

- **Let us try** to use *if instruction* to find out bigger out of two numbers.

```
#include<stdio.h>
void main(void)
{
int first, second;
scanf( " %d %d ", &first, &second);
if ( first > second )
    printf ( "First number is bigger\n");
}
```

- On the other hand, if we give input values as

10 20

- No output message will appear on the screen.

Decision Making Statements

- if statement - Example

– Let us try to use *if instruction* to find out bigger out of two numbers.

- So in order to get proper output message for all combinations of inputs, the program can be extended as:

```
#include<stdio.h>
void main(void)
{
int first, second;
scanf( " %d %d ", &first, &second);
if ( first > second )
    printf ( "First number is bigger\n");
if (second > first)
    printf ( "Second number is bigger\n");
if (first == second)
    printf ( "Both numbers are equal\n");
}
```

Decision Making Statements

- **if else** statement
 - Many times the program requires to take a specific action if a particular condition is true **and another action** if the condition is not true (i.e., false).
 - **Such a situation can be effectively implemented using if else instruction.**

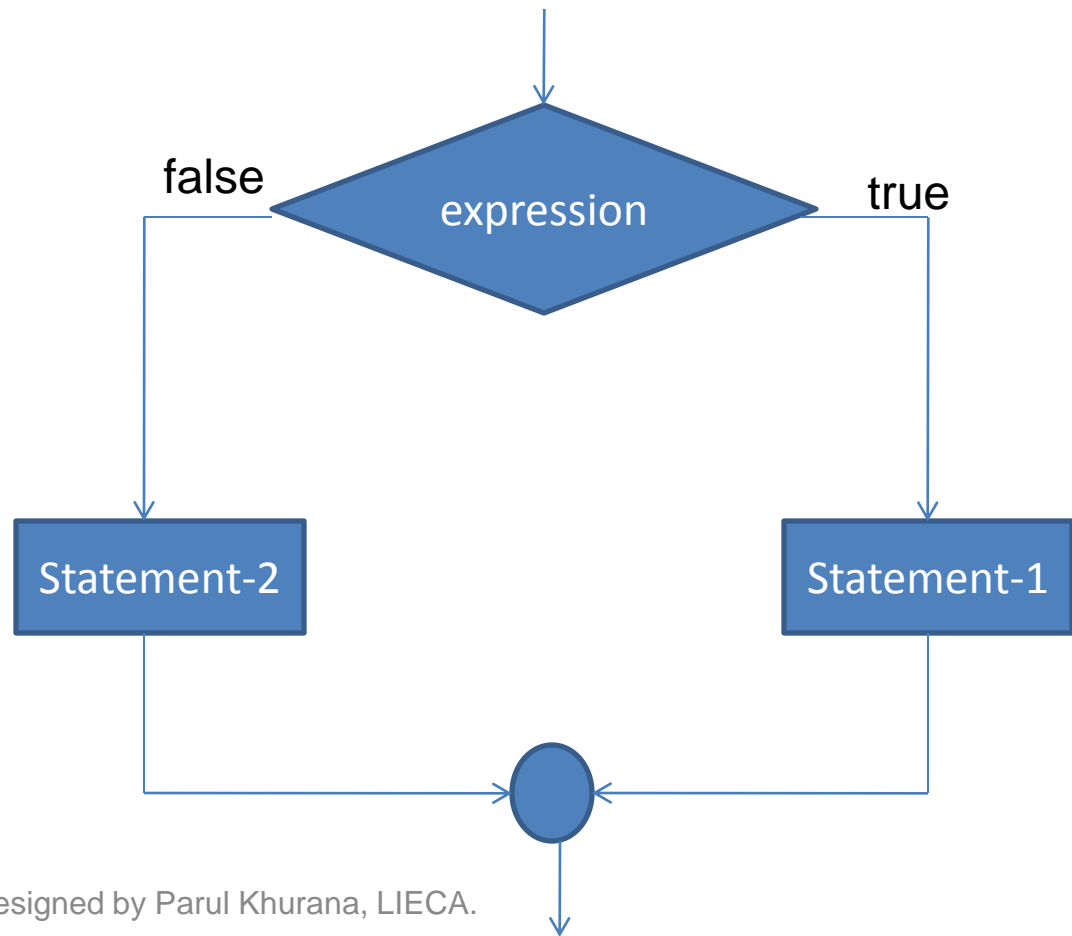
Decision Making Statements

- **if else** statement
 - **The general form of an if instruction** which includes the **else clause is:**
 - **if (expr) statement1 else statement2**
 - » If the **expr** has a **nonzero value** (i.e., if **expr** is **true**) , then **statement1** will be executed otherwise (i.e., if the **expr** is **false**), then the **statement2** will be executed.

Decision Making Statements

- Code of *if else* statement and Logic flow control

```
:  
If ( expression )  
{  
    statement -1  
}  
else  
{  
    statement -2  
}  
:  
:
```



Decision Making Statements

- **if else** statement - Example

- **Let us write** a simple program using *if-else instruction* to test whether given two numbers are equal or not.

```
#include<stdio.h>
void main(void)
{
int first, second;
scanf( " %d %d ", &first, &second);
if ( first == second )
    printf ( "Both numbers are equal\n");
else
    printf ( "Numbers are not equal\n");
}
```

- In the program, if the input value of first and second number is equal, then the statement **Both numbers are equal** will be displayed.

ECA. be displayed.

Decision Making Statements

- Nesting if else statements
 - It is possible to nest (i.e., embed) if - else statements, **one within another**. There are **several different forms** that nested if - else statements can take.
 - **The most general form of two-layer nesting is:**

```
if e1  if e2 s1
      else s2
else   if e3 s3
      else e4
```

- where **e1**, **e2** and **e3** represent logical expressions and **s1**, **s2**, **s3** and **s4** represent statements.

Decision Making Statements

- Nesting if else statements
 - Now, one complete if - else statement will be executed **if e1 is nonzero (true)**, and another complete if - else statement will be executed **if e1 is zero (false)**.
 - *It is, of course, possible that s1, s2, s3 and s4 will contain other if - else statements. We would then have multilayer nesting.*

```
if e1 if e2 s1
else s2
else if e3 s3
else e4
```

Decision Making Statements

- Nesting if else statements - SEVERAL FORMS

```
if e1 s1
```

```
else if e2 s2
```

```
if e1 s1
```

```
else if e2 s2
```

```
else e3
```

```
if e1 if e2 s1
```

```
else s2
```

```
else s3
```

```
if e1 if e2 s1
```

```
else s2
```

- In the **first three cases** the **association between the else clauses** and their corresponding expressions is **straightforward**.
- In the **last case**, however it is not clear which expression (**e1 or e2**) is associated with the else clause.
 - The answer is e2. The rule is that the **else clause is always associated with the closest preceding unmatched (i.e., else -less) if**. This is suggested by the indentation, though indentation itself is not the deciding factor.

Decision Making Statements

- **switch** statement
 - The **switch statement** causes a **particular group of statements to be chosen** from several available groups.
 - The selection is based upon the current value of an **expression which is included within the switch statement.**
 - **The general form of the switch statement is**
 - **switch (expression) statement**

Decision Making Statements

- **switch** statement
 - **The general form of the switch statement is**
 - **switch (expression) statement**
 - **where** expression results in **an integer value**. Note that **expression may also be of type** char, since *individual characters* have equivalent integer values.
 - the embedded statement is **generally a compound statement** that specifies alternate courses of action. Each alternative is expressed as a **group of one or more individual statements** within the overall embedded statement.

Decision Making Statements

- **switch** statement
 - **The general form of the switch statement is**
 - **switch (expression) statement**
 - ❑ **for each alternative, the first statement within the group must be preceded by one or more case labels (also called case prefixes).**
 - ❑ **the case labels identify the different groups of statements (i.e., the different alternatives) and distinguish them from one another.**
 - ❑ **the case labels must therefore be unique within a given switch statement.**



Decision Making Statements

- **switch** statement
 - **In** general terms, each group of statements is written as:
 - **or, when multiple case labels are required,**
 - where expression 1, expression 2,, expression m represent constant, integer-valued expressions.

case expression :

statement 1

statement 2

.....

statement n

case expression 1 :

case expression 2 :

.....

case expression m :

statement 1

statement 2

.....

statement n



Decision Making Statements

- Code of *switch case* statement

```
:
switch ( expression )
{
    case val - 1:
        statement -1;
        break;
    case val - 2:
        statement -2;
        break;
    case val - 3:
        statement -3;
        break;
    case val - n:
        statement -n;
        break;
    default:
        statement -d;
}
```

Decision Making Statements

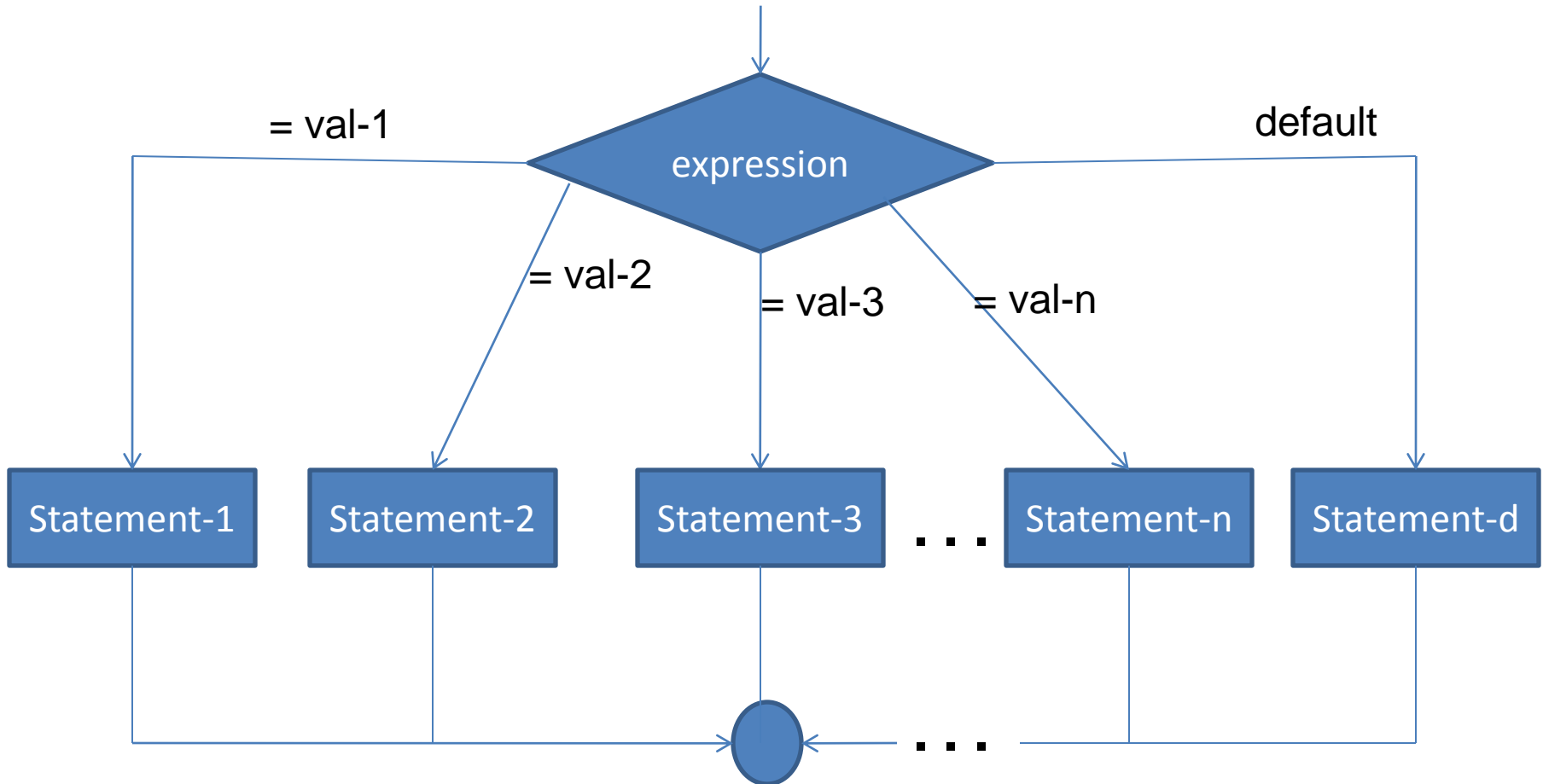
- **switch** statement - Code Explanation
 - If **expression** takes any value from **val-1, val-2, val-3,, val-n**, the **control is transferred to that appropriate case.**
 - In each case, the statements are executed and **then break statement transfers the control out of switch statement.**
 - If no **break statement** is used following a case, **except the last one in the absence of default keyword**, the control will fall through to the **next case.**

Decision Making Statements

- **switch** statement - Code Explanation
 - If the value of the expression does not match any of the case values, **control goes to the default keyword**, which is usually *at the end of the switch statement*.
 - The **use of the default keyword can be of a great convenience**.
 - If **there is no default keyword**, the **whole switch statement simply terminates** when there is no match.

Decision Making Statements

- Logic flow control of *switch case* statement- SIMPLIFIED VIEW



Decision Making Statements

- **switch case** statement - Example

– **A simple switch statement** is illustrated below. In this example, choice is assumed to be a char-type variable.

```
switch ( choice )  
{  
    case 'R':  
        printf("Red");  
        break;  
    case 'W':  
        printf("White");  
        break;  
    default:  
        printf(" Error ");  
}
```

- Thus, Red will be displayed if choice represents R, White will be displayed if choice represents W.
- **Default group will generate an Error message if none of the case labels matches the original expression.**

Decision Making Statements

- **switch case** statement - Example

– **A simple switch statement** is illustrated below. In this example, choice is assumed to be a char-type variable.

```
switch ( choice )  
{  
    case 'R':  
        printf("Red");  
        break;  
    case 'W':  
        printf("White");  
        break;  
    default:  
        printf(" Error ");  
}
```

- Note that each of first two groups ends with the break statement.
- The break statement causes control to be transferred out of the switch statement, thus preventing more than one group of statements from being executed.

Decision Making Statements

- **switch case** statement - Example

– **A simple switch statement** is illustrated below. In this example, choice is assumed to be a char-type variable.

```
switch ( choice )  
{  
    case 'R':  
        printf("Red");  
        break;  
    case 'W':  
        printf("White");  
        break;  
    default:  
        printf(" Error ");  
}
```

- One of the labeled groups of statements within the switch statement may be labeled default.
- This group will be selected if none of the case labels matches the value of the expression.

Decision Making Statements

- **switch case** statement - Example
 - **A simple switch statement** is illustrated below. In this example, choice is assumed to be a char-type variable.

```
switch ( choice )  
{  
    case 'R':  
        printf("Red");  
        break;  
    case 'W':  
        printf("White");  
        break;  
    default:  
        printf(" Error ");  
}
```

- The default group may appear anywhere within the switch statement - it need not necessarily be placed at the end.
- If none of the case labels matches the value of the expression and the default group is not present, then no action will be taken by the switch statement.

Practice Questions

- Determine the final value of x in the following cases:

```
int x = 3;  
if( x == 2 );  
    x = 0;  
if( x == 3 )  
    x ++;  
else x += 2;
```

```
int x = 5;  
int y = 2;  
char op = '*';  
switch (op)  
{  
    default : x += 1;  
    case '+' : x += y;  
    case '-' : x -= y;  
}
```