

Technical Informatics I

Control Structures (Selection) if-else if-else

by
Dr. Fatimah
Faculty of Mechanical Engineering
fatimahd@ump.edu.my



Technical Informatics 1: Dr Fatimah

Control Structures (Selection)

- Aims
 - Introduce students to logical expressions and relational operators
 - Introduce students to Control Structures (Selection): `if`, `else if`, `else`
- Expected Outcomes
 - Students are able to construct simple C programs involving appropriate application of relational structures
 - Students are able to construct simple C programs that can implement selection control structures `if`, `else if`, `else`
- References
 - Harry H. Cheng, 2010. C for Engineers and Scientists: An Interpretive Approach, McGraw Hill



Content

- Logical Expressions
- Relational Structures
- Selection Structures: If/else if/else
- Selection Structures: Nested If/else if/else
- Examples
- Conclusion



Control structures

- There are 3 control structures for C programs:

1. Sequence

- Each statement is executed sequentially (as seen in the previous lectures)

2. Selection

- One statement is *selected* over another depending on a Selection
 - If, else if, else & switch
 - If $\text{var1} > 10$, do *this...*, else do *that...*

3. Repetition

- Statements are *repeatedly* executed until it meets a certain *condition*
 - for, while, do-while loops



Control structures

- Up to now, we have worked on Control Structures (Sequence)
- In this lecture, we are going to learn how to use Control Structure (Selection)
- The way this lecture will be structured is by starting off with an example that is familiar to you (Sequential) and slowly adding building blocks to include `if`, `else if`, `else` (Selection)

Control Structure – Sequence (Review)

- **Example 1:**

Write a program that calculates and prints out the resultant pressure, p :

$$p(t) = 4(t-3)+20$$

The equation p is a function of time, t , where t is an input by the user.

Control Structure – Sequence (Review)

- **Example 1: Control structure - sequence**

```
1  /* Example for control structure - sequence*/
2  /* Calculates the force p based on the time input by the user*/
3  /* Prints out the resultant force p*/
4
5  #include <stdio.h>
6
7  - int main(){
8      /*declaration of variables*/
9      double t, p;
10     /*prompts user input for time t*/
11     printf("Enter time t:\n");
12     scanf("%lf",&t);
13     /*equation for external force, p(t)*/
14     p = 4*(t-3) + 20;
15     /*prints out resultant p*/
16     printf("The external force p(%.3f)=%.3f\n",t,p);
17     return 0;
18 }
19
```

```
>ch -u "L6-example1.c"
Enter time t:
4.23
The external force p(4.230)=24.920
>Exit code: 0
```

Logical Expressions

Operator	Logical	Description
	0	FALSE
	1	TRUE
&&	AND	If both operands are non-zero (TRUE), then the condition becomes TRUE
	OR	If one of the operands is non-zero (TRUE), then the condition becomes TRUE
!	NOT	It reverses the logical state of the operand. <ul style="list-style-type: none">• If the operand is TRUE, the logical NOT operator will make it FALSE• If the operand is FALSE, the logical NOT operator will make it TRUE

Logical Expressions

Expression	Logical Outcome
$3 > 4$	0
$3 < 4$	0
$3 == 4$	0
$3 != 4$	1
$3.0 == 3$	1

Logical Expressions

A	B	A && B	A B	!A	!B
0	0	0	0	1	1
0	1	0	1	1	0
1	1	1	1	0	0
1	0	0	1	0	1

Relational Operators

Operations	Associativity
::	Left to right
() []	Right to left
<u>function_name()</u>	Left to right
. ->	Right to left
\ ! ` ++ -- + - *	Left to right
&(type) <u>sizeof</u>	Left to right
* / % .* ./	Left to right
+ -	Left to right
<< >>	Left to right
< <= > >=	Left to right
== !=	Left to right
&	Left to right
^	Left to right
	Left to right
&&	Left to right
^^	Left to right
	Left to right
?:	Right to left
= += -= *= /=	Right to left
%= = <<= >>=	Right to left
,	Right to left

Test your understanding

- $3 < 8$ result is _____
- $3 > 8$ result is _____
- $4 \leq 4$ result is _____
- $6 \geq 4$ result is _____
- $2 == 2$ result is _____
- $2 == 7$ result is _____
- $8.0 == 8$ result is _____
- $9 != 3$ result is _____
- $(4 < 7) \parallel (3 != 8)$ result is _____
- $(4 > 7) \&\& (3 != 8)$ result is _____



Control structures

- There are 3 control structures for C programs:

1. Sequence

- Each statement is executed sequentially (as seen in the previous lectures)

2. Selection

- One statement is *selected* over another depending on a Selection
 - If, else if, else & switch
 - If $\text{var1} > 10$, do *this...*, else do *that...*

3. Repetition

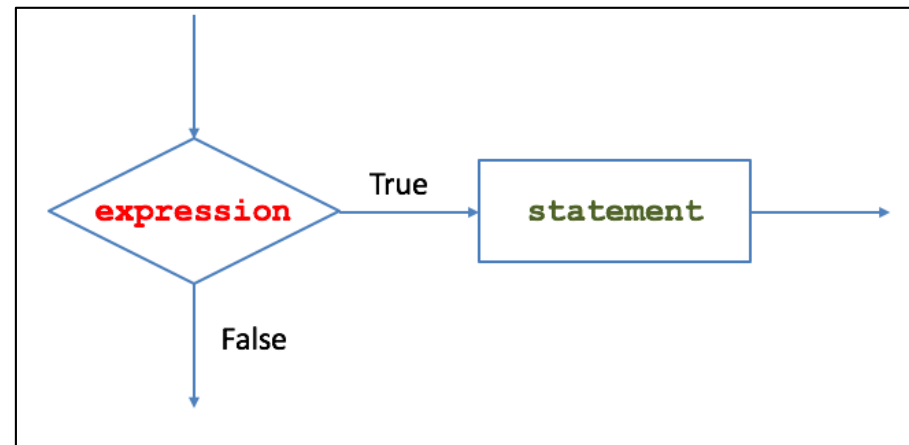
- Statements are *repeatedly* executed until it meets a certain *condition*
 - for, while, do-while loops



Flowchart of an if statement

- If the **expression** is satisfied, the **statement** will be implemented
- The syntax for an if-statement is as follows:

```
if (expression) {  
    statement  
}
```



Control Structure (Selection): if

- **Example 2:**

Write a program that calculates and prints out the resultant pressure, p:

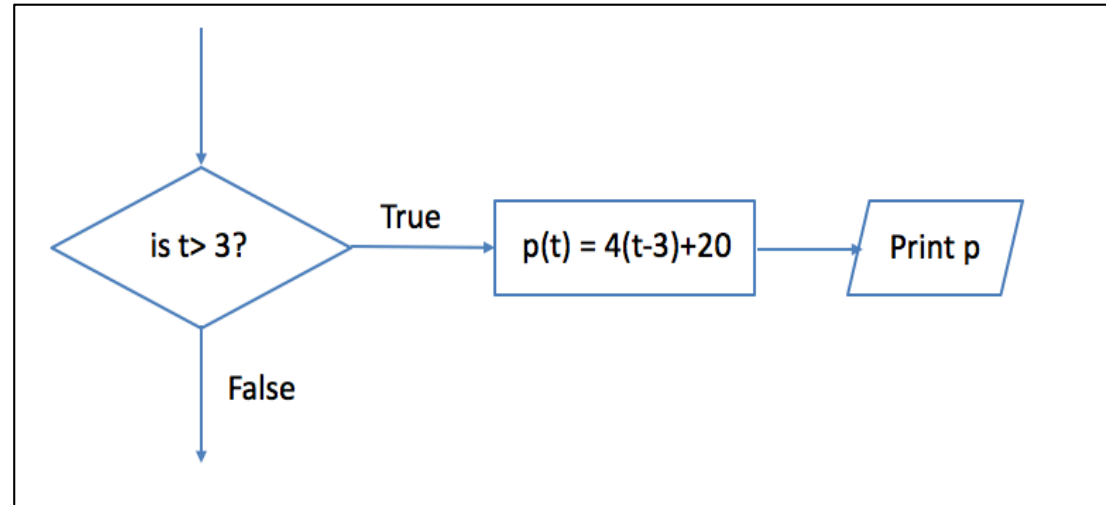
$$p(t) = 4(t-3)+20 \quad t > 3$$

The equation p is a function of time, t, where t is an input by the user.

Control Structure (Selection): `if`

- **Example 2**

```
if (t>3) {  
    p = 4*(t-3) + 20;  
    printf("p = %f",p);  
}
```



Control Structure (Selection): `if`

- Example 2

```
1  /* Example 2*/
2  /* Calculates the force p based on the time input by the user*/
3  /* Prints out the resultant force p*/
4
5  #include <stdio.h>
6
7  -int main() {
8      /*declaration of variables*/
9      double t, p;
10     /*prompts user input for time t*/
11     printf("Enter time t:\n");
12     scanf("%lf",&t);
13
14     /*is t>3?*/
15     -if(t>3){
16         p = 4*(t-3) + 20;
17         /*prints out resultant p*/
18         printf("The external force p(%.3f)=%.3f\n",t,p);
19     }
20
21     return 0;
22 }
```

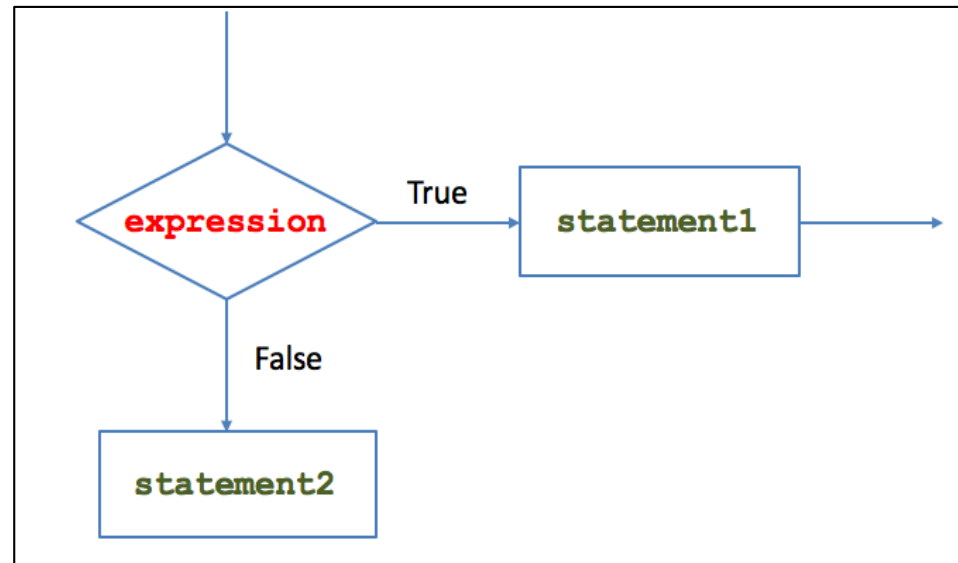
```
>ch -u "L6-example2.c"
Enter time t:
3.1
The external force p(3.100)=20.400
>Exit code: 0
```

```
>ch -u "L6-example2.c"
Enter time t:
3
The external force p(3.000)=0.000
>Exit code: 0
```


Flowchart of an if-else statement

- If **expression** is TRUE, then **statement1** is executed
- Else, if the **expression** is FALSE, then **statement2** is executed.

```
if(expression) {  
    statement1  
} else {  
    statement2  
}
```



Control Structure (Selection): `if-else`

- **Example 3:**

Write a program that calculates and prints out the resultant pressure, p :

$$p(t) = 4(t-3)+20 \quad t \geq 0$$

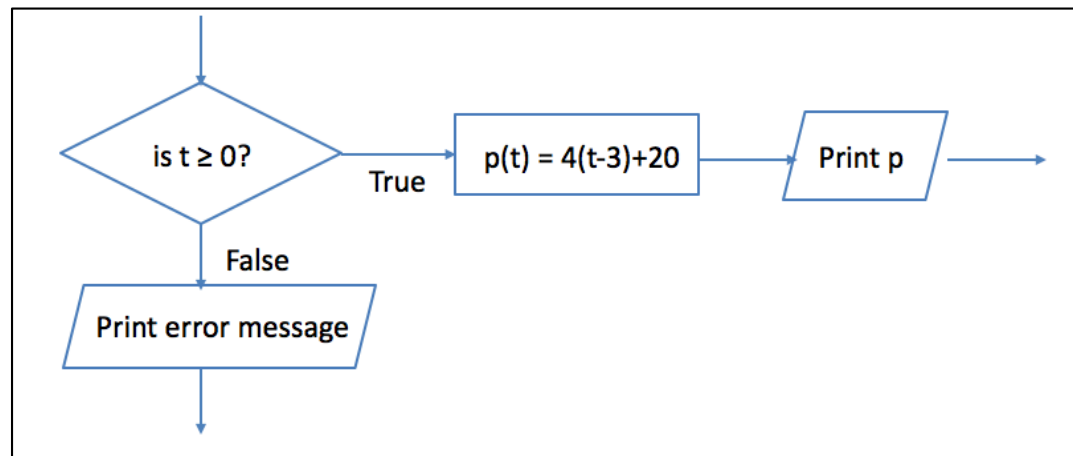
The equation p is a function of time, t , where t is an input by the user.

If the user inputs a negative value of t ($t < 0$) then an error message is printed out.

Control Structure (Selection): `if-else`

- **Example 3:**

```
if(t>=0) {  
    p = 4*(t-3) + 20;  
    printf("p = %f",p);  
} else {  
    printf("Error");  
}
```



Control Structure (Selection): `if-else`

- **Example 3:**

```
1  /* Example 3:control structure - if/else statement*/
2  /* Calculates the force p based on the time input by the user*/
3  /* Prints out the resultant force p*/
4
5  #include <stdio.h>
6
7  -int main(){
8      /*declaration of variables*/
9      double t, p;
10     /*prompts user input for time t*/
11     printf("Enter time t:\n");
12     scanf("%lf",&t);
13
14     /*is t>=0?*/
15     -if(t>=0){
16         p=4*(t-3)+20;
17         /*prints out resultant p*/
18         printf("The external force p(%.3f)=%.3f\n",t,p);
19     }
20     /*else if t<0*/
21     -else{
22         printf("Error: input t is negative\n");
23     }
24
25     return 0;
26 }
```

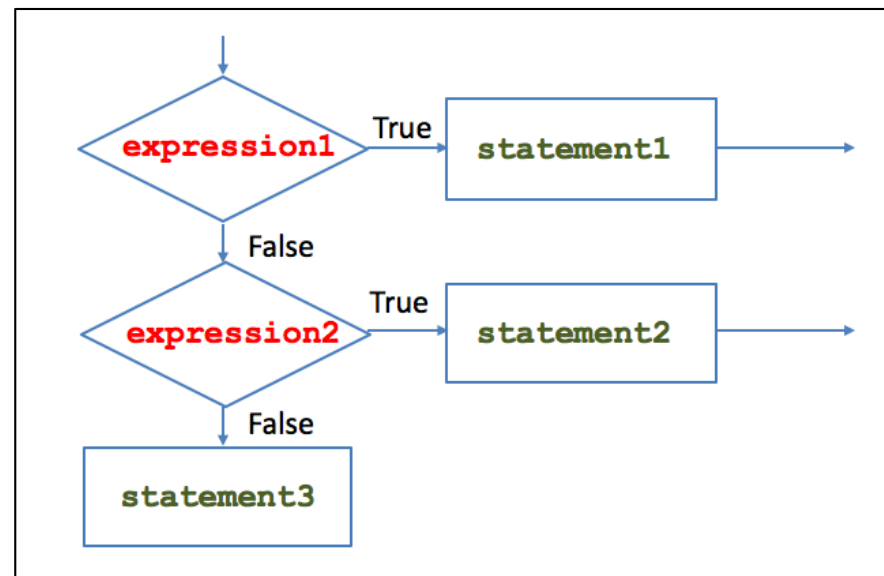
```
>ch -u "L6-example3.c"
Enter time t:
0.1
The external force p(0.100)=8.400
>Exit code: 0
```

```
>ch -u "L6-example3.c"
Enter time t:
-2.31
Error: input t is negative
>Exit code: 0
```

Flowchart for an if-else if-else statement

- If **expression1** is TRUE, then **statement1** is executed
- Else, if **expression1** is FALSE, but **expression2** is TRUE then **statement2** is executed.
- Else, if **expression1** and **expression2** are FALSE then **statement3** is executed.
- Semantically, the syntax of the `else-if` statement is an extension of the previous `if-else` statement.

```
if(expression1) {  
    statement1  
} else if(expression2) {  
    statement2  
} else {  
    statement3  
}
```



Control Structure (Selection):

if-else if - else

- **Example 4a:**

Write a program that calculates and prints out the resultant pressure, p:

$$p(t) = 20 \qquad 0 \leq t < 3$$

$$p(t) = 4(t-3)+30 \qquad t \geq 3$$

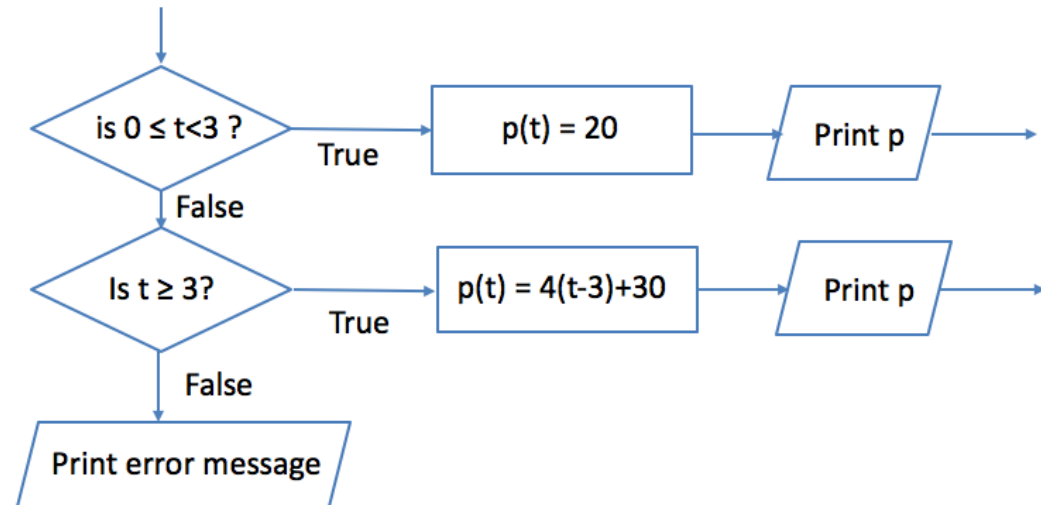
The equation p is a function of time, t, where t is an input by the user.

If the user inputs a negative value of t ($t < 0$) then an error message is printed out.

Control Structure (Selection): if-else if - else

- **Example 4a:**

```
if (t >= 0 && t < 3) {  
    p = 20;  
    printf("p = %f", p);  
} else if (t >= 3) {  
    p = 4 * (t - 3) + 30;  
    printf("p = %f", p);  
} else {  
    printf("Error");  
}
```



Control Structure (Selection): if-else if - else

- **Example 4a:**

```

1  /* Example 4:control structure - if/else if/else statement*/
2  /* Calculates the force p based on the time input by the user*/
3  /* Prints out the resultant force p*/
4
5  #include <stdio.h>
6
7  - int main(){
8      /*declaration of variables*/
9      double t, p;
10     /*prompts user input for time t*/
11     printf("Enter time t:\n");
12     scanf("%lf",&t);
13
14     /*is 0<=t<3?*/
15     - if(t>=0 && t<3){
16         p = 20;
17         /*prints out resultant p*/
18         printf("The external force p(%.3f)=%.3f\n",t,p);
19     }
20     /*is t>=3?*/
21     - else if(t>=3){
22         p=4*(t-3)+30;
23         /*prints out resultant p*/
24         printf("The external force p(%.3f)=%.3f\n",t,p);
25     }
26     /*else if t<0*/
27     - else{
28         printf("Error: input t is negative\n");
29     }
30
31     return 0;
32 }
33

```

```

>ch -u "L6-example4.c"
Enter time t:
1
The external force p(1.000)=20.000
>Exit code: 0

```

```

>ch -u "L6-example4.c"
Enter time t:
3
The external force p(3.000)=30.000
>Exit code: 0

```

```

>ch -u "L6-example4.c"
Enter time t:
-5
Error: input t is negative
>Exit code: 0

```


Nested if-else if-else statements

- An example of the syntax for the nested else-if statement is as follows:

```
if(expression1) {  
    statement1  
    if(expression2) {  
        statement2  
    } else if(expression3) {  
        statement3  
    } else {  
        statement4  
    }  
}
```

- If **expression1** is TRUE, then the inner if-else if-else statements (in the blue box) will be evaluated.

- Else, if **expression1** is FALSE then the inner if-else if-else statements (in the blue box) will not be evaluated.

Nested if-else if-else statements

- **Example 4b:**

Write a program that calculates and prints out the resultant pressure, p:

$$p(t) = 20 \qquad 0 \leq t < 3$$

$$p(t) = 4(t-3)+30 \qquad t \geq 3$$

The equation p is a function of time, t, where t is an input by the user.

If the user inputs a negative value of t ($t < 0$) then an error message is printed out.

Note: This example is the same as **Example 4a** but this time, we're going to implement a nested if-else if statement



Nested if-else if-else statements

- **Example 4b:**

```
if (t >= 0) {  
    if (t >= 0 && t < 3) {  
        p = 20;  
    } else if (t >= 3) {  
        p = 4 * (t - 3) + 30;  
    }  
} else {  
    printf("Error");  
}
```

- If **t >= 0** is TRUE, then the inner if-else statements (in the blue box) will be evaluated.
- Else, if **t >= 0** is FALSE then an error message will be printed out.

Nested if-else if-else statements

- Example 4b

```

1  /* Example 4b:nested if/else if/else statement*/
2  /* Calculates the force p based on the time input by the user*/
3  /* Prints out the resultant force p*/
4
5  #include <stdio.h>
6
7  int main(){
8      /*declaration of variables*/
9      double t, p;
10     /*prompts user input for time t*/
11     printf("Enter time t:\n");
12     scanf("%lf",&t);
13
14
15     if(t>=0){
16         /*is 0<=t<3?*/
17         if(t>=0 && t<3){
18             p = 20;
19         }
20         /*is t>=3?*/
21         else if(t>=3){
22             p=4*(t-3)+30;
23         }
24         /*prints out resultant p*/
25         printf("The external force p(%.3f)=%.3f\n",t,p);
26     }
27     /*else if t<0*/
28     else{
29         printf("Error: input t is negative\n");
30     }
31
32     return 0;
33 }

```

```

>ch -u "L6-example4b.c"
Enter time t:
1
The external force p(1.000)=20.000
>Exit code: 0

```

```

>ch -u "L6-example4b.c"
Enter time t:
3
The external force p(3.000)=30.000
>Exit code: 0

```

```

>ch -u "L6-example4b.c"
Enter time t:
-5
Error: input t is negative
>Exit code: 0

```

Conclusion

```
if (expression) statement;
```

```
if (expression) {  
    Block of statements;  
}
```

```
if (expression) {  
    Block of statements;  
} else {  
    Block of statements;  
}
```

Conclusion

```
if (expression) {  
    Block of statements;  
} else if(expression) {  
    Block of statements;  
} else {  
    Block of statements;  
}
```

Technical Informatics I

Lecture 4

Dr Fatimah



Technical Informatics 1: Dr Fatimah