

# CONTROL STRUCTURES IN C

# INTRODUCTION

---

- Any statement that alters or changes the sequential flow of control is called as a control structure.
- There are mainly three types of control structures:
  - Branching - Conditional structures (No repeated execution)
  - Loops – Repetitive execution
  - Jump statements – Transfer from one point to another point/ may skip statements of execution

# BRANCHING STATEMENTS

- if statement:

- Syntax:

- if(condition)

- {

- statements;

- }

- if the condition is true, it will execute the statements within the “if” block

# BRANCHING STATEMENTS

- if –else statement:

- Syntax:

```
if(condition)
{
statements;
}
else
{
statements;
}
```

- if the condition is true, it will execute the statements within the “if” block.
- if the condition is false, it will execute the statements within the else block.

# BRANCHING STATEMENTS

## ➤ nested if-else:

- An if-else statement can be inserted within an if or an else block.
- Syntax:

```
if(condition)
{
    if(condition)
    {
    }
    else
    {
    }
}
else
{
    if(condition)
    {
    }
    else
    {
    }
}
```



# BRANCHING STATEMENTS

## ➤ else-if ladder

➤ More than one conditions

➤ Syntax:

```
if( condition 1 )
```

```
{  
}
```

```
else if ( condition 2 )
```

```
{  
}
```

```
else if ( condition 3 )
```

```
{  
}
```

```
else
```

```
{  
}
```

➤ Note that else block is optional.

# QUICK EXERCISE

- Find the output:

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int x=9;
```

```
if(x<8)
```

```
printf("Hello");
```

```
printf("Bye!");
```

```
}
```

Output:

Bye

# QUICK EXERCISE

➤ Find the output:

```
#include<stdio.h>
void main()
{
int x=10;
if(x=9)
{
printf("Hello\n");
printf("Bye!");
}
else
printf("Welcome");
}
```

Output:

Hello

Bye!



# QUICK EXERCISE

- Find the output:

```
#include<stdio.h>
void main()
{
int x=10;
if(x<=9);
printf("Hello\n");
printf("Bye!\n");
printf("Welcome");
}
```

**Output:**

Hello

Bye!

Welcome

# QUICK EXERCISE

➤ Find the output:

```
#include<stdio.h>
void main()
{
int x=10;
if(x>=9)
printf("Hello");
printf("Bye!");
else
printf("Welcome");
}
```

**Output:**

**Error**

**\*\*\*An else must immediately follow an if**

# QUICK EXERCISE

➤ Find the output:

```
#include<stdio.h>
void main()
{
    int x=10, y=6;
    if(x<=9)
    {
        if(y<5)
            printf("Hello");
        else
            printf("Hello");
    }
    else
    {
        if(y >5)
            printf("Bye!");
        else
            printf("Welcome");
    }
}
```

Output:

Bye!

# BRANCHING STATEMENTS

- Points to note:
  - If the { } are missing, only the first statement will be considered as a part of if/else.
  - DO NOT put a ; after the if condition i.e,  
if(condition); **X**
  - *if* can exist without an *else*. But *else* cannot exist without a corresponding *if*.
  - No condition needs to be specified for an *else*.
  - In case of else-if ladder, once the condition becomes true, no further conditions will be checked. Control is transferred outside the else-if

# BRANCHING STATEMENTS

## ➤ switch:

- Multiple outcomes depending upon the value of an expression/variable.

- Syntax:

```
switch(variable/expression)
{
case value1: statements;
             break;
case value2: statements;
             break;
case value_n: statements;
             break;
default: statements;
}
```



# BRANCHING STATEMENTS

- Points to note:
  - default case is optional.
  - The case values must be a constant only.
  - The case values can only be an *int* or a *char* constant.
  - break statement takes the control outside the switch. In the absence of break, AFTER it finds the first match, all the cases following it will be executed.
  - Duplicate case values will raise a compiler error.

# QUICK EXERCISE

- Find the output when the value of x = 5 and x=1:

switch (x)

{

case 7: printf("7");

case 5: printf("5");

case 3: printf("3");

break;

case 1: printf("1");

default: printf("Default");

}

Output:

When x=5,

53

When x=1,

1Default

# HOW TO DECIDE WHICH STRUCTURE TO USE?

---

- Given a problem statement, check :
  1. How many outcomes are possible?
    - If only two outcomes, use an if-else.
    - If multiple outcomes, go for else-if ladder or switch.
  2. In case of multiple outcomes, check:
    - If outcomes depends upon conditions (true/false type). Then opt for else-if ladder.
    - Or values such as an int/char then opt for a switch.

# QUICK EXERCISE

- Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer (Out of 50 each). Calculate percentage and grade according to following:
  - Percentage between 90-100% : Grade A
  - Percentage between 80-90% : Grade B
  - Percentage between 70-80 % : Grade C
  - Percentage between 60-70% : Grade D
  - Percentage between 40-50% : Grade E
  - Percentage < 40% : Grade F



# QUICK EXERCISE

- Given the rectangular co-ordinates  $(x,y)$  , find the polar co-ordinates of a point  $(r, \theta)$  using the formula below. Check if the angle  $(\theta)$  is more than 60 degrees or not.
- To convert from Cartesian Coordinates  $(x,y)$  to Polar Coordinates  $(r,\theta)$ :  
$$r = \sqrt{ x^2 + y^2 }$$
$$\theta = \tan^{-1} ( y / x )$$



# QUICK EXERCISE

- Print the largest of three numbers using nested if-else.
- Write a menu-driven program to Calculate the area of the following shapes depending upon the user choice. If user enters any other number, print “Invalid choice”.
  1. Circle
  2. Square
  3. Rectangle
  4. Triangle