# **C Switch**

## Switch Statement

Instead of writing **many**if..else statements, you can use the switch statement.

The switch statement selects one of many code blocks to be executed:

### **Syntax**

switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}

This is how it works:

* The switch expression is evaluated once
* The value of the expression is compared with the values of each case
* If there is a match, the associated block of code is executed
* The break statement breaks out of the switch block and stops the execution
* The default statement is optional, and specifies some code to run if there is no case match

The example below uses the weekday number to calculate the weekday name:

### **Example**

int day = 4;  
  
switch (day) {  
  case 1:  
    printf("Monday");  
    break;  
  case 2:   
    printf("Tuesday");  
    break;  
  case 3:  
    printf("Wednesday");  
    break;  
  case 4:  
    printf("Thursday");  
    break;  
  case 5:  
    printf("Friday");  
    break;  
  case 6:  
    printf("Saturday");  
    break;  
  case 7:  
    printf("Sunday");  
    break;  
}

# **C While Loop**

Loops can execute a block of code as long as a specified condition is reached.

Loops are handy because they save time, reduce errors, and they make code more readable.

int i = 0;  
  
while (i< 5)

{  
  printf("%d\n", i);  
  i++;  
}

int i = 0;  
do {  
  printf("%d\n", i);  
  i++;  
}while (i< 5);

# **C For Loop**

int i;  
  
for (i = 0; i< 5; i++)

{  
  printf("%d\n", i);  
}

## Nested Loops

### **Example**

int i, j;  
  
// Outer loop  
for (i = 1; i<= 2; ++i)

{  
  printf("Outer: %d\n", i);  // Executes 2 times  
 // Inner loop  
  for (j = 1; j <= 3; ++j)

{  
    printf(" Inner: %d\n", j);  // Executes 6 times (2 \* 3)  
}  
}