# Control Statements in C with Example Programs

## 1. Conditional Statements

Conditional statements are used to perform different actions based on different conditions.

* **Example: if Statement**

The if statement allows a program to execute a block of code only if a specified condition is true. It’s used for decision-making.

**Code:**

#include <stdio.h>  
int main() {  
 int num = 10;  
 if (num > 0) {  
 printf("Number is positive\n");  
 }  
 return 0;  
}

* **Example: if-else Statement**

The if-else statement executes one block of code if a condition is true, and a different block of code if the condition is false.

**Code:**

#include <stdio.h>  
int main() {  
 int num = -5;  
 if (num > 0) {  
 printf("Number is positive\n");  
 } else {  
 printf("Number is non-positive\n");  
 }  
 return 0;  
}

* **Example: else if Ladder**

Used when multiple conditions need to be checked. It provides a way to chain several conditions.

**Code:**

#include <stdio.h>  
int main() {  
 int num = 0;  
 if (num > 0) {  
 printf("Number is positive\n");  
 } else if (num < 0) {  
 printf("Number is negative\n");  
 } else {  
 printf("Number is zero\n");  
 }  
 return 0;  
}

* **Example: switch Statement**

The switch statement is used to evaluate a single expression against multiple cases and execute the corresponding block of code.

**Code:**

#include <stdio.h>  
int main() {  
 int day = 3;  
 switch (day) {  
 case 1:  
 printf("Monday\n");  
 break;  
 case 2:  
 printf("Tuesday\n");  
 break;  
 case 3:  
 printf("Wednesday\n");  
 break;  
 default:  
 printf("Other day\n");  
 }  
 return 0;  
}

## 2. Looping Statements

Looping statements are used to execute a block of code repeatedly.

* **Example: for Loop**

A for loop is used when the number of iterations is known beforehand. It repeats a block of code for a specified number of times.

**Code:**

#include <stdio.h>  
int main() {  
 for (int i = 1; i <= 5; i++) {  
 printf("Value: %d\n", i);  
 }  
 return 0;  
}

* **Example: while Loop**

A while loop repeats a block of code as long as a specified condition remains true. It’s often used when the number of iterations is not known.

**Code:**

#include <stdio.h>  
int main() {  
 int i = 1;  
 while (i <= 5) {  
 printf("Value: %d\n", i);  
 i++;  
 }  
 return 0;  
}

* **Example: do-while Loop**

The do-while loop is similar to the while loop, but it guarantees that the block of code will execute at least once, even if the condition is false.

**Code:**

#include <stdio.h>  
int main() {  
 int i = 1;  
 do {  
 printf("Value: %d\n", i);  
 i++;  
 } while (i <= 5);  
 return 0;  
}

## 3. Jump Statements

Jump statements are used to alter the flow of control unconditionally.

* **Example: break Statement**

The break statement is used to exit a loop or switch statement prematurely. It ends the loop or switch and resumes execution after it.

**Code:**

#include <stdio.h>  
int main() {  
 for (int i = 1; i <= 5; i++) {  
 if (i == 3) {  
 break;  
 }  
 printf("Value: %d\n", i);  
 }  
 return 0;  
}

* **Example: continue Statement**

The continue statement is used to skip the current iteration of a loop and move to the next iteration.

**Code:**

#include <stdio.h>  
int main() {  
 for (int i = 1; i <= 5; i++) {  
 if (i == 3) {  
 continue;  
 }  
 printf("Value: %d\n", i);  
 }  
 return 0;  
}