Decision Control Statement

- In decision control statements (C if else and nested if), group of statements are executed when condition is true. If condition is false, then else part statements are executed.
- There are 3 types of decision making control statements in C language. They are,

1. if statements
2. if else statements
3. nested if statements

“If”, “else” and “nested if” decision control statements in C:

- Syntax for each C decision control statements are given in below table with description.

Decision control statements	Syntax	Description

if	if (condition)
	{ Statements; }
		In these type of statements, if condition is true, then respective block of code is executed.

if…else	if (condition)
	{ Statement1; Statement2; }
	else
	{ Statement3; Statement4; }
		In these type of statements, group of statements are executed when condition is true. If condition is false, then else part statements are executed.

nested if	if (condition1){ Statement1; }
	else if(condition2)
	{ Statement2; }
	else Statement 3;
		If condition 1 is false, then condition 2 is checked and statements are executed if it is true. If condition 2 also gets failure, then else part is executed.

Example program for if statement in C:

In “if” control statement, respective block of code is executed when condition is true.

```
int main()
{
  int m=40,n=40;
  if (m == n)
  {
  printf("m and n are equal");
  }
}
```

Output:

```
m and n are equal
```

Example program for if else statement in C:

In C if else control statement, group of statements are executed when condition is true. If condition is false, then else part statements are executed.

```
#include <stdio.h>
int main()
{
  int m=40,n=20;
  if (m == n)
  {
  printf("m and n are equal");
  }
  else
  {
  printf("m and n are not equal");
  }

}
```

Output:

```
m and n are not equal
```

Example program for nested if statement in C:

- In "nested if" control statement, if condition 1 is false, then condition 2 is checked and statements are executed if it is true.
- If condition 2 also gets failure, then else part is executed.

```
#include <stdio.h>
int main()
{
  int m=40,n=20;
  if (m>n) {
 printf("m is greater than n");
  }
  else if(m<n) {
  printf("m is less than n");
  }
  else {
  printf("m is equal to n");
  }
}
```

Output:

```
m is greater than n
```