SESSION-3

Control Statements in C:

In C, programs can choose which part of the code to execute based on some condition. This ability is called **decision making** and the statements used for it are called **control/conditional statements.** These statements evaluate one or more conditions and make the decision whether to execute a block of code or not.

Example:

```
#include <stdio.h>
int main() {
 // Number of people in the audience
 int num = 100;
 // Conditional code inside decision making statement
 if (num > 50) printf("Start the show");
return 0;
}
```

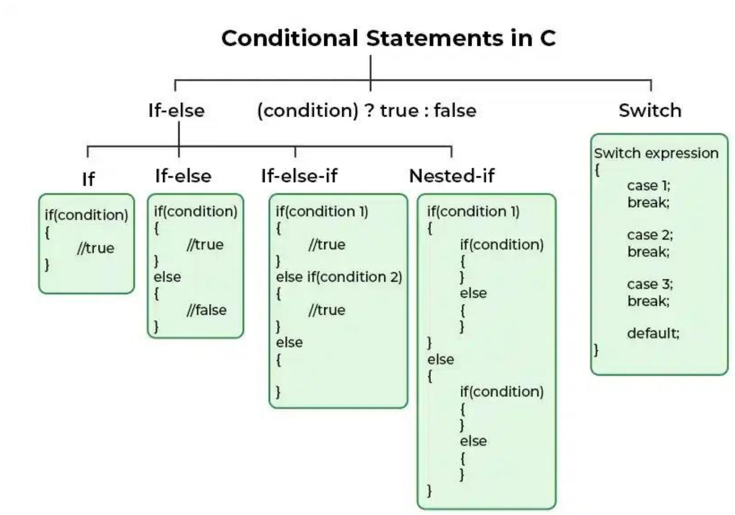
Output:

```
Start the show
```

In the above program, the show only starts when the number of people is greater than **50**. It is specified in the **if statement** (a type of conditional statement) as a condition **(num > 50)**. You can decrease the value of **num** to less than **50** and try re-running the code.

Types of Conditional Statements in C:

In the above program, we have used if statement, but there are many different types of conditional statements available in C language:



```
if(condition)
{
    //true
}
```

```
if(condition)
{
    //true
}
else
{
    //false
}
```

```
if(condition 1)
{
    //true
}
else if(condition 2)
{
    //true
}
else
{

}
```

```
if(condition 1)
{
    if(condition)
    {
    }
    else
    {
    }
}
else
{
    if(condition)
    {
    }
    else
    {
    }
}
```

```
Switch expression
{
    case 1;
    break;

    case 2;
    break;

    case 3;
    break;

    default;
}
```

SESSION-3

1. If in C:

The if statement is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e. if a certain condition is true then a block of statements is executed otherwise not.
A **condition** is any expression that evaluates to either a true or false (or values convertible to true or false).

Example:

```
#include <stdio.h>
int main() {
int age = 20;
// If statement
if (age >= 18) {
printf("Eligible for vote");
}
return 0;
}
```

Output:

```
Eligible for vote
```

The expression inside **() parenthesis** is the **condition** and set of statements inside **{} braces** is its **body**. If the condition is true, only then the body will be executed.

2. If-else in C:

The **if** statement alone tells us that if a condition is true, it will execute a block of statements and if the condition is false, it won't. But what if we want to do something else when the condition is false? Here comes the C **else** statement. We can use the **else** statement with the **if** statement to execute a block of code when the condition is false. The if-else statement consists of two blocks, one for false expression and one for true expression.

Example:

```
#include <stdio.h>
int main() {
int age = 10;
if (age >= 18)
printf("Eligible for vote");
else
printf("Not Eligible for vote");
return 0;
}
```

SESSION-3

Output

```
Not Eligible for vote
```

3. Nested if-else in C

A nested if in C is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement. Yes, C allow us to nested if statements within if statements, i.e, we can place an if statement inside another if statement.

Example:

```
#include <stdio.h>
int main(){
int age = 11;
if (age >= 18) {
if (age >= 60)
printf("Eligible to vote (Senior Citizen)\n");
else
printf("Eligible for vote\n");
}
else {
printf("Not eligible to vote (Under 18)\n");
if (age >= 13)
printf("teenager\n");
else
printf("not a teenager\n");
}
return 0;
}
```

Output

```
Not eligible to vote (Under 18)
not a teenager
```

4. if-else-if Ladder in C:

The if else if statements are used when the user has to decide among multiple options. The C if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. if-else-if ladder is similar to the switch statement.

Example:

SESSION-3

```
#include <stdio.h>
int main() {
int i = 20;
// If else ladder with three conditions
if (i == 10) printf("Not Eligible");
else if (i == 15) printf("wait for three years");
else if (i == 20) printf("You can vote");
else printf("Not a valid age");
return 0;
}
```

Output:

```
You can vote
```

5. Switch Statement in C:

The switch case statement is an alternative to the if else if ladder that can be used to execute the conditional code based on the value of the variable specified in the switch statement. The switch block consists of cases to be executed based on the value of the switch variable.

Example:

```
#include <stdio.h>
int main() {
// variable to be used in switch statement
int var = 18;
// declaring switch cases
switch (var) {
case 15: printf("You are a kid");
break;
case 18: printf("Eligible for vote");
break;
default: printf("Default Case is executed");
break;}
return 0;
}
```

Output:

```
Eligible for vote
```

SESSION-3

Problems on Leetcode on Operators:

1. Power of Two
2. Two Sum