Control Statements - I

hitbullseye

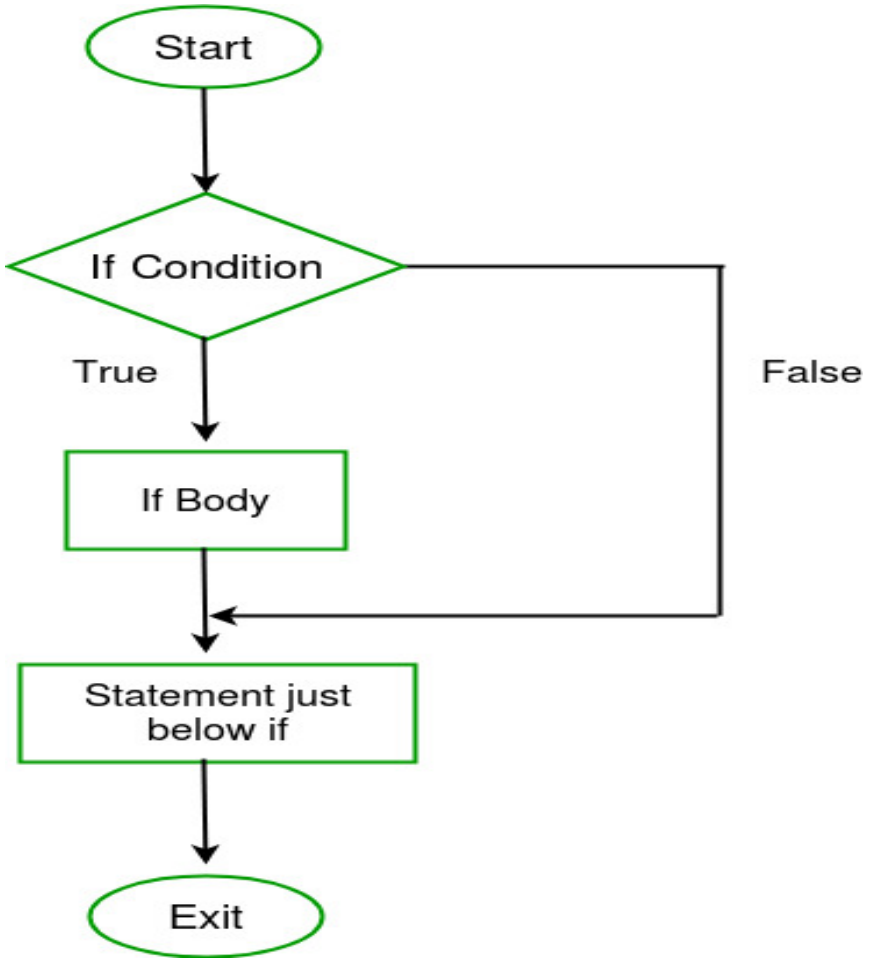
If Statements

If Statement:

- if statement is the most simple decision-making statement.
- It is used to decide whether a block of statements will be executed or not.
- If the condition is true, then a certain block of code will execute.

The Syntax is:

```
if (condition x)
{
Statement1;
Statement 2;
}
```



Program:

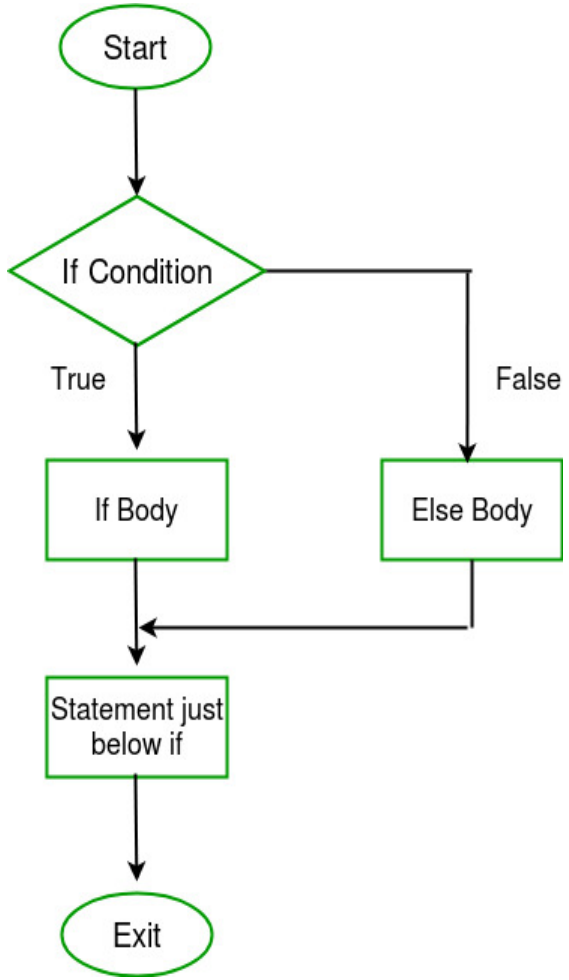
```
#include <stdio.h>
int main()
{
int num = 20;
    int num1= 22;
    if (num<num1)
    {
        printf ("num is less than num1");
    }
    return 0;
}
```

Output: num is less than num1

If else statement

if else statement:

- If-else statement make decisions according to the given conditions.
- If the given condition is true, then the statements inside the body of “if” will Execute , and the statements inside the body of “else” are ignored.
- if the condition is false, then the statements inside the body of “if” are ignored, and the statements inside the “else” are executed.



The Syntax is:

```
if (condition a)
{ Statement a; Statement b; }
Else
{ Statement x; Statement y; }
```

Program:

```
#include<stdio.h>
int main()
{
    int x = 200;
    if (x < 150) {
        printf (“x is smaller than 150");
    }
    else {
        printf (“x is greater than 150");
    }
    return 0;
}
```

Output: x is greater than 150

If else if statement

if else if statement:

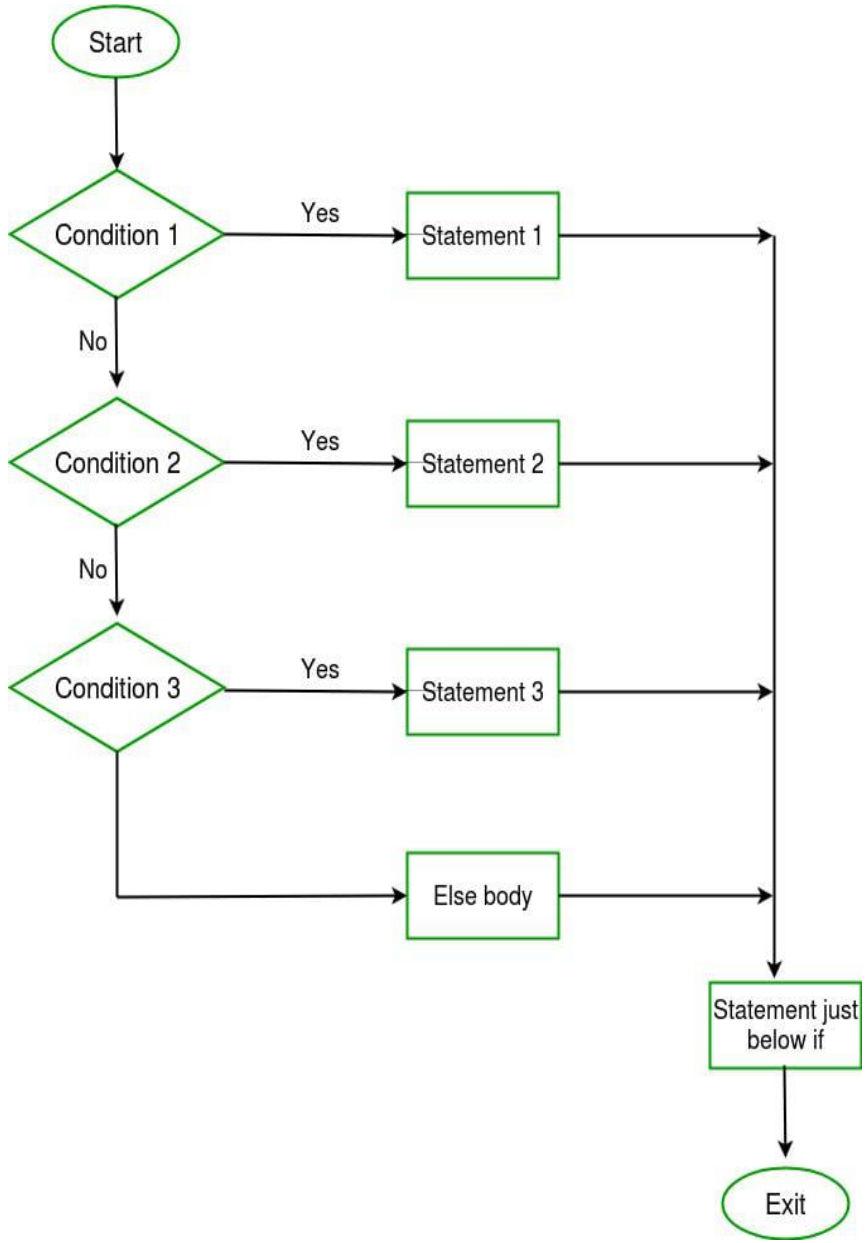
- In this user has multiple options to select.
- When the statements are executed from the top down. If one of the condition is true, then the statement written in that block will be executed, and the rest of the C else-if ladder is ignored. If none of the conditions is true, then the final else statement will be executed.

Syntax of if-else-if

```
if (condition)
Statement;//if condition is true then this block will execute
else if (condition)
statement;
else
Statement; }
```

Program:

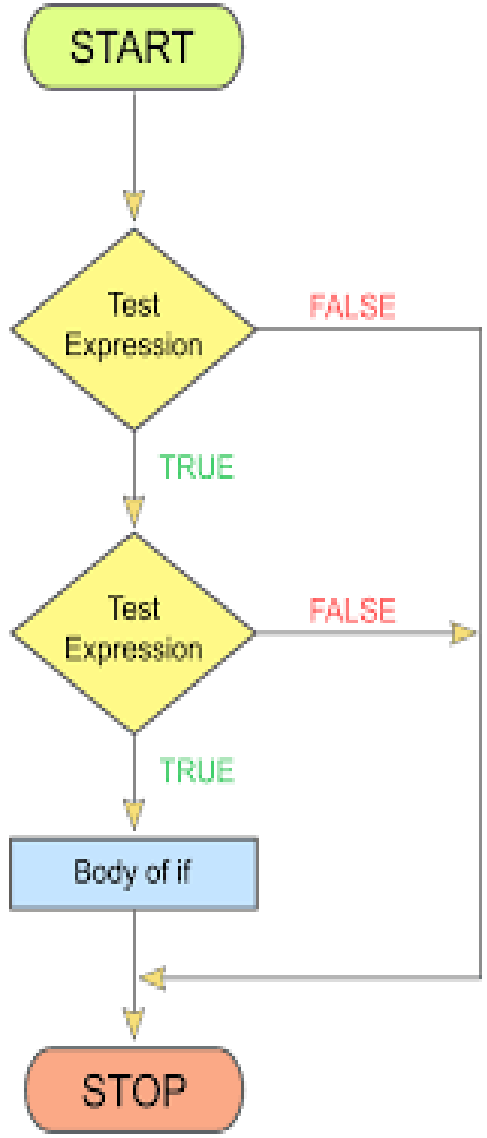
```
#include<stdio.h>
int main()
{ int i = 200;
 if (i == 100)
        printf("i is 100");
    else if (i == 150)
        printf("i is 150");
    else if (i == 200)
        printf("i is 200");
    else
        printf("i is not present");
} Output: i is 200.
```



nested if statement

nested if statement:

- nested if statements mean an if statement inside another if statement.
- nested if statement is helpful if you want to check condition inside a condition.



Syntax of nested if:

```
if (condition1)
{ // Executes when condition1 is true
if (condition2)
{ // Executes when condition2 is true
 } }
```

Program:

```
#include<stdio.h>
int main()
{
    int i = 100;
    if (i == 100) {
                if (i < 150)
            printf("i is smaller than 150”);
          if (i < 120)
            printf("i is smaller than 120”);
         else
            printf("i is greater than 150");
    } return 0;}
```

Output: i is smaller than 150
 i is smaller than 120

Switch Statement

- Switch statement tests the value of a variable and compares it with multiple cases.
- If the match is found then block of statements will execute find in that particular case.
- Each value is called Case. Case value can be integer or character type.
- Duplicate case values are not allowed.

Switch Statement

Two keywords used in switch statement:

***Break*: This keyword is used to stop the of switch block.**

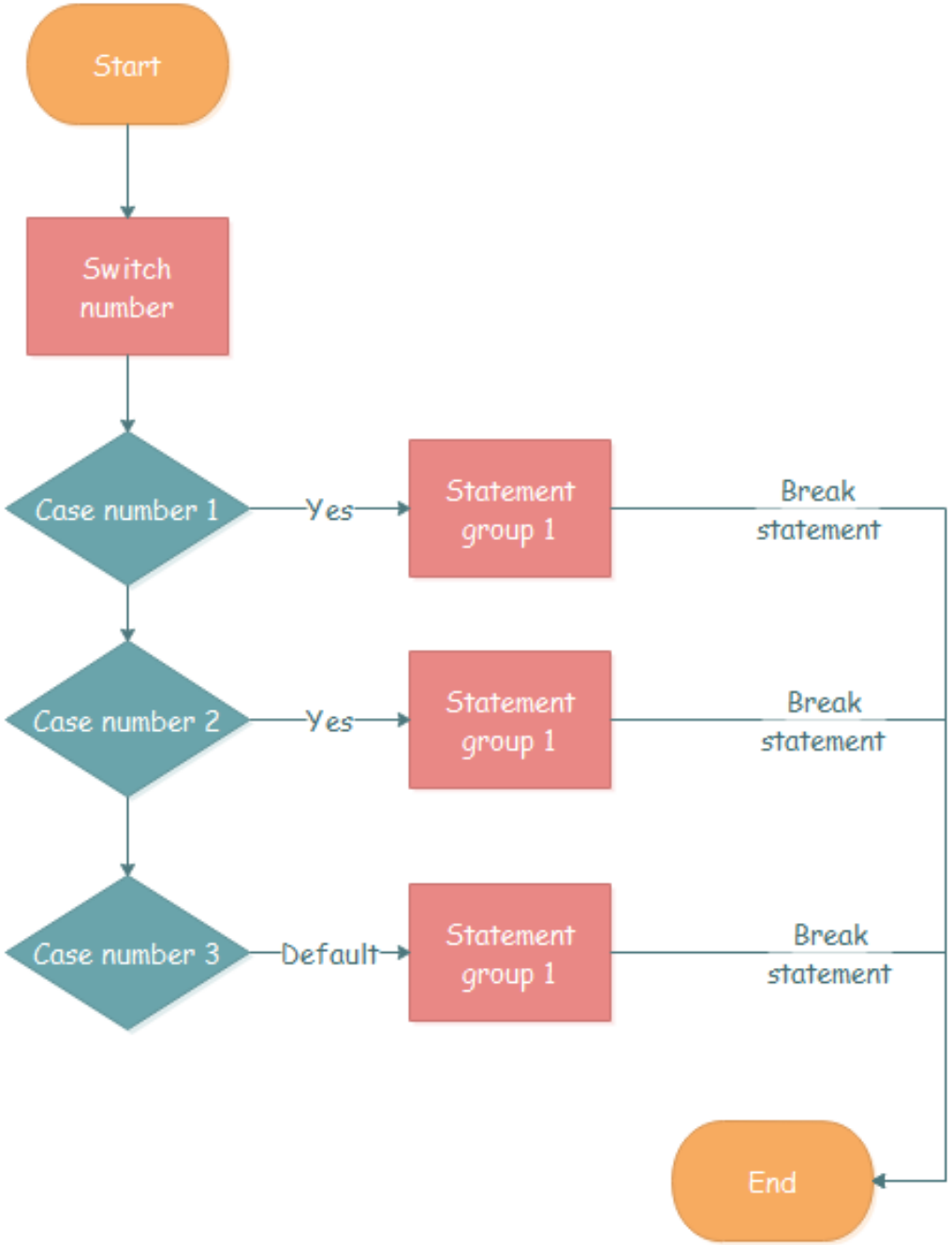
***Default:* This keyword is used to specify the set of statements to execute if there is no match with the case.**

Rules while using the switch statement:

1. One or N numbers of cases.

2. Unique values must be there.

3. Break statement must be there after every statement.



Program:

```
#include <stdio.h>
int main()
{   int a= 2;
    switch (a) {
    case 1:printf("The value is 1");
        break;
    case 2: printf("The value is 2");
        break;
    case 3:printf("The value is 3");
        break;
    default: printf("Value is 7");
        break;
    }return 0;}
```

Output is : The value is 2

Nested –Switch Statement

- Nested switch defines switch statements inside of another Switch Statements.
- The outer switch statement is the first switch statement whereas the inside switch is referred to as an inner switch statement.

Program:

```
#include <stdio.h>
int main() {
   int x = 100;
   int y = 200;
   switch (x) {
      case 100:  printf("First switch\n");
         switch (y) {
            case 200: printf("Inner switch\n"); }}
   printf("Value of x is: %d ", x);
   printf("Value of y is: %d ", y);
return 0;}
```

Output:

Output: First switch

Inner switch

Value of x is: 100 Value of y is: 200

References:

Student reference link:

https://www.javatpoint.com
https://www.tutorialspoint.com/cprogramming
https://www.tutorialspoint.com/index.htm
https://www.edrawsoft.com
https://www.studymite.com/c-programming-language/nested-switch-in-c
https://www.youtube.com/watch?v=ILbaH1gIZ2I

THANK YOU

hitbullseye