Maya G C

Mayagc2000@gmail.com

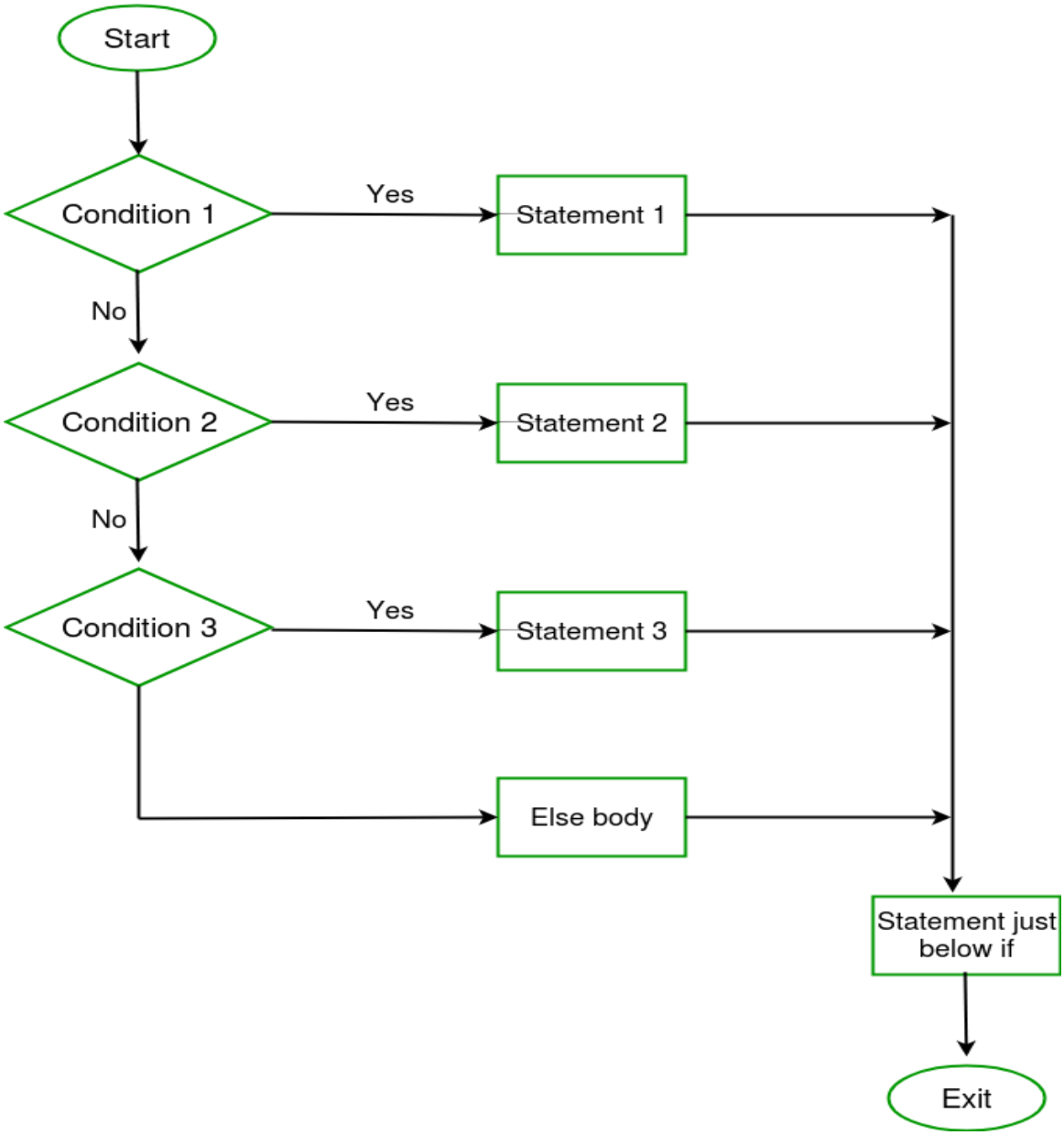
9/08/2023

Kodnest

# ASSIGNMENTS

| Parameter | If-else | Switch |
|---|---|---|
| Definition | The if and else blocks are executed depending on the condition in the if statement | The switch statement has multiple cases, and the code block corresponding to that case is executed |
| Evaluation | Used for integer, character, pointer, floating-point type, or Boolean type. | Used for character expressions and integers. |
| Testing | Tests both logical expressions and equality | Tests only equality |
| Expression | Multiple statements for multiple decisions | Single statements for multiple decisions |
| Default Execution | If the condition inside the if-statement is false, then the code block under the else condition is executed | If the condition inside switch statements does not match any of the cases, the default statement is executed. |
| Sequence of Execution | Either the code block in the if statement is executed or the code block in the else statement. | The switch case statement performs each case until a break statement is encountered or the end of the switch statement is reached. |
| Speed | If you use 'if-else' to implement several options, the speed will be slow. | If we have numerous options, the switch statement is the best solution because it executes considerably faster than the 'if-else' statement. |
| Editing | Difficult to edit nested if-else statements. | Easy to edit. |
| Values | Based on constraint | Based on user |

**FLOW DIAGRAM: IF ELSE**




**IF ELSE CONDITIONAL CONTROL CONSTRUCT:**

```
import java.io.*;

class GFG {

    public static void main(String[] args){

        int i = 20;

         if (i == 10)

            System.out.println("i is 10\n");
```

```
        else if (i == 15)
            System.out.println("i is 15\n")
        else if (i == 20)
            System.out.println("i is 20\n");
else
            System.out.println("i is not present\n");
System.out.println("Outside if-else-if");
   }
}
```

## Nested if condition :

**Nested means within. Nested if condition means if-within-if. Nested if condition comes under decision-making statement in Java. There could be infinite if conditions inside an if condition. The below syntax represents the Nested if condition.**

## Syntax :

```
if(condition ){

       if(condition){

                  if(condition){

                              ......

                  }

        }
}
```

## NESTED IF CONDITION:

```
import java.util.*;
import java.lang.*;
import java.io.*;
class GFG
{
   public static void main(String args[])
   {
      int a=10;
       int b=20;
     if(a==10){
        if(b==20){
           System.out.println("GeeksforGeeks");
        }
     }
   }
}
```

## NESTED IF ELSE CONDITION:

We saw how helpful if and else statements are, but what if we need to check for more conditions even when one condition is satisfied? In such cases, we use nested if-else statement. Nesting is the practice of enclosing several if-else statements within an if-and-else statement.

```
If (condition)
{
    Statement1;
```

}

Else

{

Statement2;

}

```
import java.util.*;
import java.lang.*;
class GFG
{
   public static void main(String args[])
   {
      int a=10;
       int b=20;
      if(a==10){
     if(b!=20){
          System.out.println("GeeksforGeeks");
        }
 else{
          System.out.println("GFG");
        }
}
}
}
```