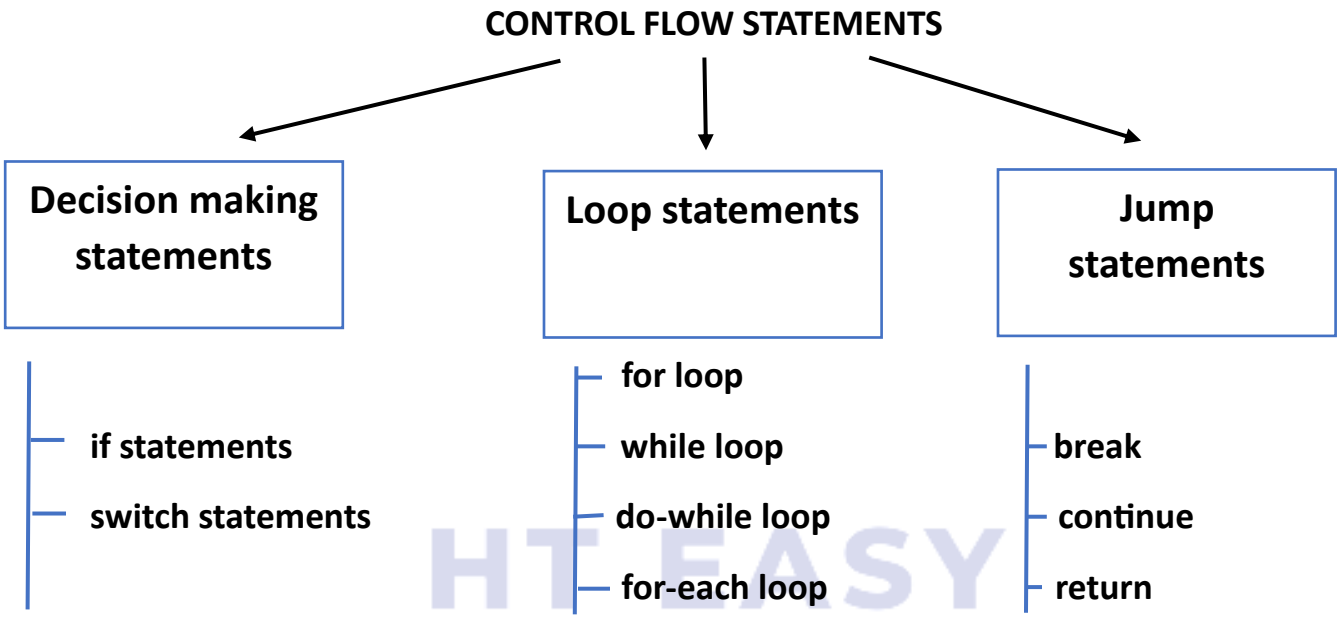
CORE JAVA

JAVA CONTROL STATEMENTS-



Decision making statements-

Decision-making statements in Java allow you to control the flow of execution based on conditions.

Basically, based on some condition your program will decide which statement to execute and which to not.

For example, we put a condition that if age is greater than 18 then print- “eligible to vote” and if less than 18 then print- “Not eligible to vote”.

There are two types of decision-making statements in Java-

1. if statements
2. switch statements

Switch Statements-

Switch statement is a conditional statement in java that allows us to perform specific operations based on the case value matching the expression value. Switch statement is an alternative to if-else if ladder statement when checking a variable against multiple possible values.

Syntax:

```
switch (expression) {
    case value1:
        // Code to be executed if expression equals value1
        break;
    case value2:
        // Code to be executed if expression equals value2
        break;
    // You can have any number of case statements
    default:
        // Code to be executed if none of the cases match
        break;
}
```

Explanation:

- switch is the keyword used to define switch statement followed by circular bracket and inside which we have to provide an expression. This expression is then checked with each of the case, if the case matches the expression, then that case will be executed. If none of the case matches the expression then the default code will be executed.
- **case**: Each case compares the expression to a specific value. If a match is found, the corresponding code block runs.
- The break statement is used to stop the execution and come out of the switch block. **[break is an optional statement]**
- **break**: It terminates/stops the switch block once a case is matched. Without break, the execution "falls through" to subsequent cases, which might lead to unwanted behavior.
- **default**: This is optional and executes if none of the cases match. Think of it like the final else in an if-else ladder.

Let's understand this with an example-

SwitchStatement1 - Notepad

File Edit Format View Help

```
import java.util.Scanner;
public class SwitchStatement1{
        public static void main(String args[])
        {
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the day:");
                int day=s.nextInt();
                switch(day)
                {
                        case 1:
                                System.out.println("Monday");
                                break;
                        case 2:
                                System.out.println("Tuesday");
                                break;
                        case 3:
                                System.out.println("Wednesday");
                                break;
                        case 4:
                                System.out.println("Thursday");
                                break;
                        case 5:
                                System.out.println("Friday");
                                break;
                        case 6:
                                System.out.println("Saturday");
                                break;
                        case 7:
                                System.out.println("Sunday");
                                break;
                        default:
                                System.out.println("Enter valid day");
                                break;
                }

        }
}
```

Output-

```
C:\Users\LENOVO\Documents\java_practical>javac SwitchStatement1.java

C:\Users\LENOVO\Documents\java_practical>java SwitchStatement1
Enter the day:
2
Tuesday

C:\Users\LENOVO\Documents\java_practical>java SwitchStatement1
Enter the day:
10
Enter valid day
```

Explanation:

- During the first run, we provided the value 2 for the variable day from the console. The switch statement checked if the value of day (the expression) matched the value of case 1 (the case value). Since it did not match, the code inside case 1 did not execute. Then, it checked if the value of day (i.e., 2) matched case 2. Since it did, case 2 was executed, and it printed "Tuesday" to the console.
- Afterward, the break statement terminated the execution, and we exited the switch statement. If there will be a statement outside the switch it will be executed, as break only terminates the execution of the switch block.
- If we hadn’t used break, the remaining cases would have also been executed.
- In the second run, we provided 10 as the value for day, and since none of the cases matched, the default case was executed.

break statement:

break is a jump statement in Java. It is used to terminate the execution of loops and switch statement.

In a switch statement, break is used to exit the switch block once a matching case is executed. Without break, the program would continue executing subsequent cases.

The break statement can be used to exit a loop early, stopping all further iterations and immediately moving to the next statement after the loop.

We will learn about break and loops in detail in the upcoming chapters.

Let's see another example of switch statement:

SwitchStatement2 - Notepad

File Edit Format View Help

```
import java.util.Scanner;
public class SwitchStatement2
{
        public static void main(String args[])
        {
                Scanner s=new Scanner(System.in);
                System.out.print("Enter first number:");
                int a=s.nextInt();
                System.out.print("Enter second number:");
                int b=s.nextInt();
                System.out.print("Enter the operation you want to perform:");
                char op=s.next().charAt(0);

                switch(op){
                        case '+':
                                System.out.println("Addition:"+(a+b));
                                break;
                        case '-':
                                System.out.println("Substraction:"+(a-b));
                                break;
                        case '*':
                                System.out.println("Multiplication:"+(a*b));
                                break;
                        case '/':
                                System.out.println("Division:"+(a/b));
                                break;
                        default:
                                System.out.println("Enter valid operation to perform");
                                break;
                }

        }
}
```

Output:

```
C:\Users\LENOVO\Documents\java_practical>javac SwitchStatement2.java

C:\Users\LENOVO\Documents\java_practical>java SwitchStatement2
Enter first number:10
Enter second number:4
Enter the operation you want to perform:+
Addition:14

C:\Users\LENOVO\Documents\java_practical>java SwitchStatement2
Enter first number:15
Enter second number:5
Enter the operation you want to perform:x
Enter valid operation to perform
```

HOMEWORK:

Question 1: Create a Java program using a switch statement where the user-entered expression is checked against cases for vowels (a, e, i, o, u). If it is not a vowel, print that it is a consonant.

Question 2: Create a Java program using a switch statement for months (Jan, Feb, March, etc.). When the user enters 1, case 1 will be executed, printing "January." Similarly, the same logic applies for other months. If the user enters a number outside the range of 1 to 12, print "Enter a valid month.

Join our growing community of tech enthusiasts! Subscribe to our YouTube channel, where we simplify programming languages in the easiest way possible. Gain a deeper understanding with our clear explanations and receive exclusive notes to enhance your learning journey. Don't miss out on valuable insights – hit that subscribe button now and embark on a programming adventure with us!

Subscribe to our channel:
https://www.youtube.com/@HTEASYLEARNING