1. **Else if statement**

Here user can decide multiple option while executing the statement from top as soon as one of the condition is true the statement associated with will execute and rest of condition are skipped. If none of the condition is true then final else statement will execute

Note :- If multiple condition are true then it will execute first condition

Program:

```
package ControlProg;

public class ElseIfProg {

	public static void main(String[] args) {

		int marks = 101;

		if(marks>=40 && marks<60)
		{
			System.out.println("Student is pass in second class");
		}
		else if(marks>=60 && marks<75)
		{
			System.out.println("Student is pass in first class");
		}
		else if(marks>=75 && marks<90)
		{
			System.out.println("Student is pass in Distinction");
		}
		else if(marks>=90 && marks<=100)
		{
			System.out.println("Student is pass in Merit");
		}
		else if(marks>100)
		{
			System.out.println("Invalid input");
		}
		else
		{
			System.out.println("Student is fail");
		}
	}
}

//40-59 ---> Second class
//60-74 ---> First class
//75-89 ---> Distinction
//90-100 ---> Merit
//<40 ---> Fail
//>100 ----> NA / Wrong input/marks
//and &
//0	&	0 = 0
```

```
//0     &     1 = 0
//1     &     0 = 0
//1     &     1 = 1

//  or |
//0     |     0 = 0
//0     |     1 = 1
//1     |     0 = 1
//1     |     1 = 1
```

2. **Nested if statement**

   It means if statement inside if statement

Program:

```
package ControlProg;

public class NestedIfProg {
//WAP to login to FB
	public static void main(String[] args) {

		String username = "admin";
		String password = "Admin@1234";
		// = ---> assignment operator
		// == ---> use for comparison
		if(username == "admin")
		{
			System.out.println("Username is correct, now enter pwd");
			if(password == "Admin@1234")
			{
				System.out.println("Login sucessful");

			}
			else
			{
				System.out.println("incorrect password entered");
			}
		}
		else
		{
			System.out.println("Username is incorrect");
		}
	}
}
```

3. **Switch Case**

The Java switch statement executes one statement from multiple conditions.

Points about Switch Statement:

There can be one or N number of case values for a switch expression.

The case value must be of switch expression type only.
The case values must be unique. In case of duplicate value, it renders compile-time error.
Each case statement can have a break statement which is optional.
The case value can have a default label which is optional.

Switch Case Syntax :

```
switch(expression){ case value1:
//code to be executed;
break; //optional case value2:
//code to be executed;
break; //optional ......
default:
code to be executed if all cases are not matched;
}
```

Program:

```
package ControlProg;

public class SwitchCaseProg1 {

	public static void main(String[] args) {

		String day = "Saturday";  //+ - / *

		switch(day)
		{
		case "Monday":
			System.out.println("Today is Monday");
			break;
		case "Tuesday":
			System.out.println("Today is Tuesday");
			break;
		case "Wednesday":
			System.out.println("Today is Wednesday");
			break;
		case "Thursday":
			System.out.println("Today is Thursday");
			break;
		case "Friday":
			System.out.println("Today is Friday");
			break;
		default:
			System.out.println("Wow, its weekend!");
		}
//		System.out.println("Switch finished");
	}

}
```