CONDITIONAL AND LOOPS

Condition:- It provide check for the statement.

1. If-else statement → Used to check the condition, it checks the Boolean condition True or False.

   Syntax :-

```
if (boolean expression True or false){
    //Body
} else{
    // Do this
}
```

   Example:-

```
public class IfElse {
    public static void main(String[] args) {
        int salary = 25400;
        if (salary> 10000) {
            salary = salary + 2000;
        }
        else {
            salary = salary + 1000;
        }

        System.out.println(salary);

    }
}
```

   Output :- 27400

2. Multiple if-else statement

   → It executes one condition from multiple statements.

   Syntax :-

```
if (condition 1){
    // code to be executed if condition 1 is true
} else if (condition 2) {
    // code to be executed if condition 2 is true
} else if (condition 3){
    // code to be executed if condition 3 is true
} else {
    // code to be executed if all conditions are false
}
```

   Example :-

```
public class MultipleIfElse {
    public static void main(String[] args) {
        int salary = 25400;
        if (salary<= 10000) {
            salary +=1000;
        }
        else if (salary <= 20000) {
            salary += 2000;
        }
        else {
            salary += 3000;
        }
        System.out.println(salary);

    }
}
```

   Output :- 28400

Loop → Loops are used to iterate a part of program several times.

1. for loop :- It is generally used when we know how many times loop will iterate.
   Syntax :-

```
for (initialization; condition; increment/decrement){
    // body
}
```

2. While Loop :- It is used when we don't know how many time the loop will iterate.
   Syntax :-

```
while (condition){
    // code to be executed
    // increment/decrement
}
```

3. do while loop :- It is used when we want to execute our statement at least one time.
   → It is called exit control loop because it checks the condition after execution of statement.
   Syntax :-

```
do{
    // code to be executed
    // update statement -> increment/decrement
}while (condition);
```

While Loop	Do while loop
→ used when no. of iteration is not fixed	→ used when we want to execute the statement at least ones
→ Entry controlled loop	→ Exit controlled loop
→ no semicolon required at the end of while (condition)	→ semicolon is required at the end of while (condition)

6/8/21 If-else conditions
Loops → while & for & do-while

7/8/21 Switch Statements + Nested case in Java.

• Switch Statements :

```
switch (expression) {
    case one:
        // code block
        break;
    case two:
        // code block
        break;
    default:
        // code block
}
```

break; → terminate the sequence

default: → default will execute when none of above does.
→ if default is not at end put break after it.

→ if break is not used then it will continue with other cases.

→ duplicate cases not allowed.

eg:
```
case one:
    // code block
    break;
case one:
    // code block
    break;
```

✗ not allowed.

New Syntax:

```
switch (expression) {
        case one -> //do this ;
        case two -> // do this;
        default -> // do this;
}
```

* `x.equals("word")` → here equals only checks value not reference.

`x == "word"` → here it checks reference

- **Nested Switch Case:**

```
switch (expression) {
        case one:
            // code block
            break;
        case two:
            switch (expression) {
                case one:
                    //code block
                    break;
                case two:
                    //code block
                    break;
                default;
                    // code block
            }
            break;
        default:
            // code block
}
```