21. Control Flow Statements in Java | Switch Exp, Switch Cases, do-while / while Loop

Switch Case

- In Switch Cases if break not applied, at any case and that case gets choosen, All the code below it gets executed till a break statement is encountered.

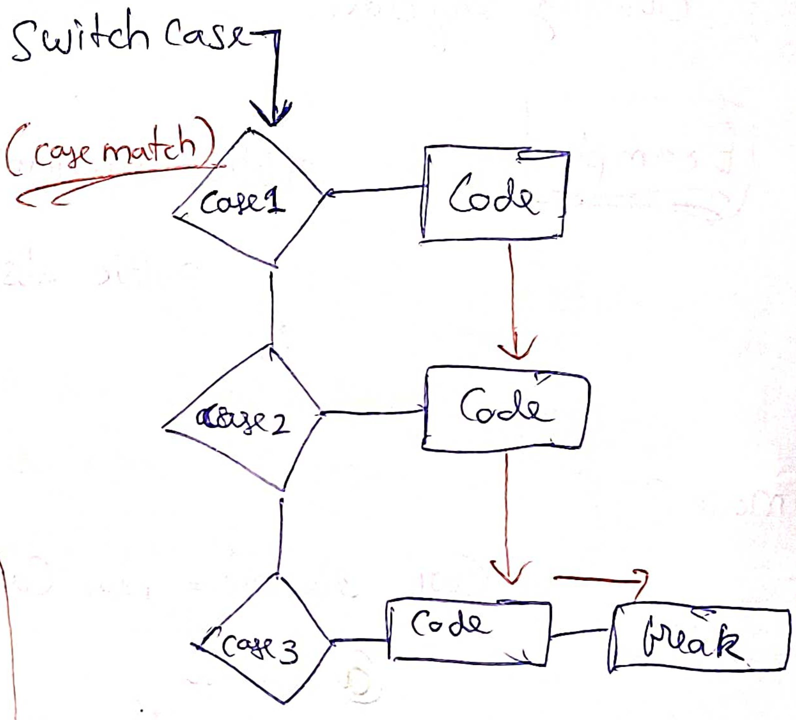
(Example)

```
int a=10;

switch (a) {

  case 10:
        -- code --
  case 20:
        --- code --
  case 30:
        --- code --
  default:
        -----
        break.
}
```

All the code from this till any break or end of default section gets executed.



* We can merge cases together. Even write the "," separated.

```
switch (a) {

  case 10:
  case 20:
  ------
      Code section --
}
```

OR

```
switch (a){

  case 10, 20 :

}
```

* We need not always need to write default block in the end. But if we are writing default block in the middle, we must apply break statement.

```
exp = 2
switch ( exp) {

    case 1 :
            ....code.....
            break;
                                    Satisfying this use case.
    default :
            .....code.....  [Code executed]

    case 20:
            ... code ...  [code executed]

    case 10:
            ... code ...  [ code executed].
};
```

while , do-while Loop.

Only Difference between while and do-while is that do while will ~~get~~ executed once even if the expression is False

```
int val = 10;

do {
      val++;  ⟶ val becomes 11

}
while (val < 10);
    ↳ Here next time it breaks.
```

Switch case few rules.

* Case value has to be a LITERAL / CONSTANT. Cannot be variable
* Do not need to handle all use cases. You can even skip default.
* Nested Switch statement is possible.
* Supported datatypes to use in switch(datatype):
  1) 4 Primitive types: int, short, byte, char
  2) Wrapper Classes: Integer, Short, Byte, Character
  3) Enum
  4) String

Nested Switch Case

Example in Code

```
int age = 10;
String month = "Nov";

switch (age){

   case 10,20,40:

      switch(month){

         ---- Few Cases

      };

   ↓ Few more cases

}
```

Return Statements are not possible within Switch Case

```
int age = 1
String val = Switch(age){

   case 1:
      return "1";

};
```

↓

{Compile ERROR will be thrown}

(Switch Expression) [Java 12 onwards]

In Switch Case ~~[illegible]~~ we saw that returning from switch statements is not allowed.

Switch Expression allows us to do so

1. (Case N →) Using this.

Example

```
int age=10;

String val = switch (age) {

        case 10 → "Ten";
        default → "None";

    }.
```

① Break Statement Not required since its returning from here.

② All possible use cases need to be handled since we are returning something

2. Yield Statement

Problem with case ~~N~~ → we can only return, we cannot write a block of code inside it and return.

This problem is solved by Yield Statement

```
String val= switch (exp) {

    case exp : {_ _ _ _ _
          _ _ _ Code block
          yield "20";
    }

}.
```