Flow and Decision Control
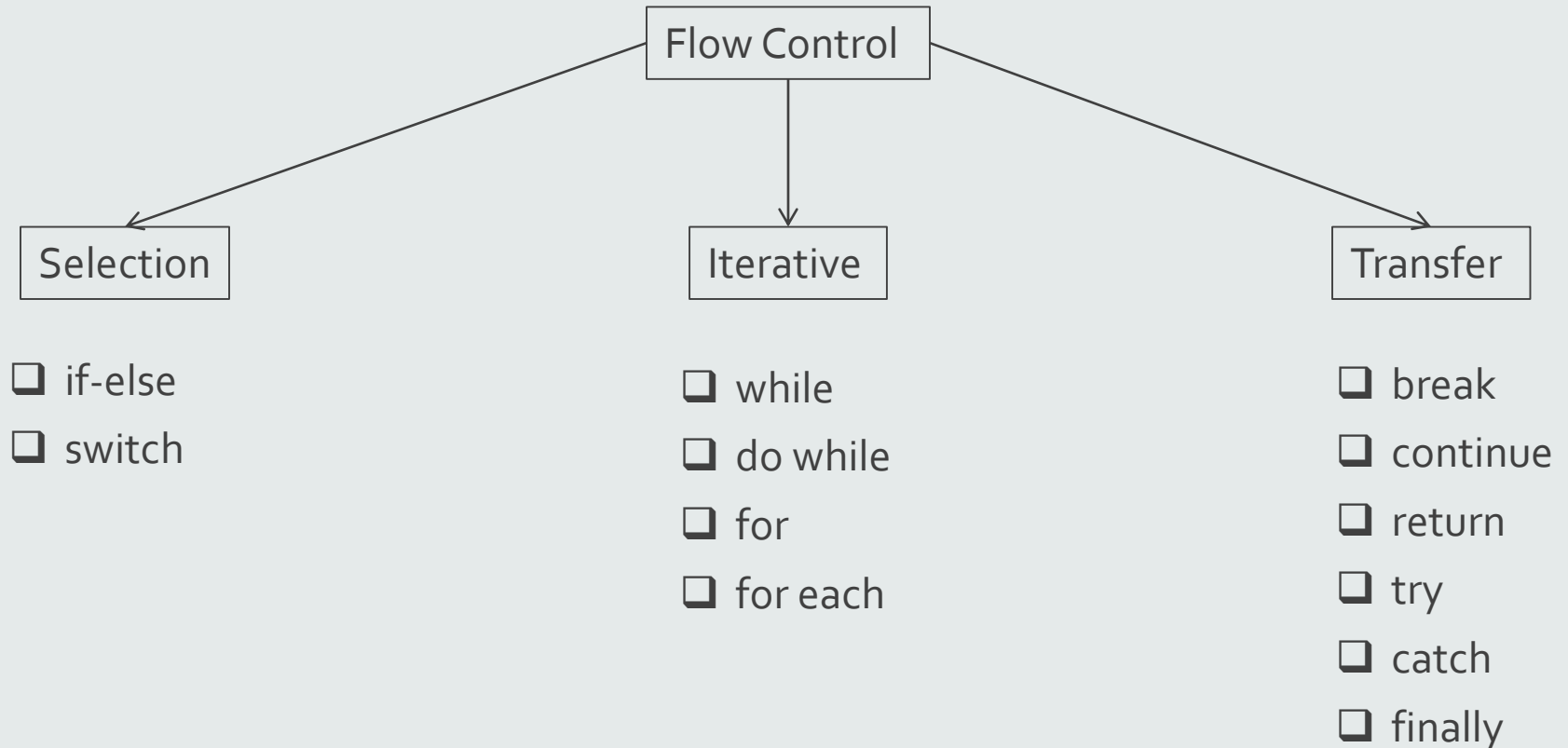
Ram Sharma

Flow Control

- It is the order in which statements will be executed at runtime.

```
                    ┌──────────────┐
                    │ Flow Control │
                    └──────────────┘
          ┌────────────────┼────────────────┐
          ▼                ▼                 ▼
   ┌───────────┐    ┌───────────┐     ┌───────────┐
   │ Selection │    │ Iterative │     │ Transfer  │
   └───────────┘    └───────────┘     └───────────┘
```

Selection	Iterative	Transfer
❑ if-else	❑ while	❑ break
❑ switch	❑ do while	❑ continue
	❑ for	❑ return
	❑ for each	❑ try
		❑ catch
		❑ finally

if else

- Syntax

```
if(var1){
        when var1 is true
} else{
        when var1 is false
}
```

- var1 should always be a boolean value.
- if you do not provide any curly braces, the very next statement followed by if or else will be considered.
- You can also use if – else if – else
- You can also do nesting of if-else.

switch statement

- Syntax
  ```
  switch(var1){
  case label1:
          Action 1;
  case label2:
          Action 2;
          break;
  default:
          Default Action;
  }
  ```

- It is one of the many situation.

- Each label should be a constant.

- Duplicate labels are not allowed.

- break is used to stop fall through inside switch.

- if none of the labels are matched, it will come to default.

while and do-while loop

- Syntax of while
  ```
  while(var1){
          Action when var1 is true
  }
  ```

- Syntax of do-while
  ```
  do{
          Action when var1 is true
  } while(var1)
  ```

- var1 must be of boolean type.

- Mostly used when no of iteration is not known in advance.

- If curly braces not provided, only next one statement is considered.

- With do-while, the loop will be entered at-least once.

for loop

- Syntax

    for(initialization; condition check; increment/decrement){

        Loop body

    }

- Initialization will be executed only once at the beginning.

- Condition check can be any java expression whose result is a boolean value, empty is considered as true.

- increment/decrement can be any valid java statement.

- Process:
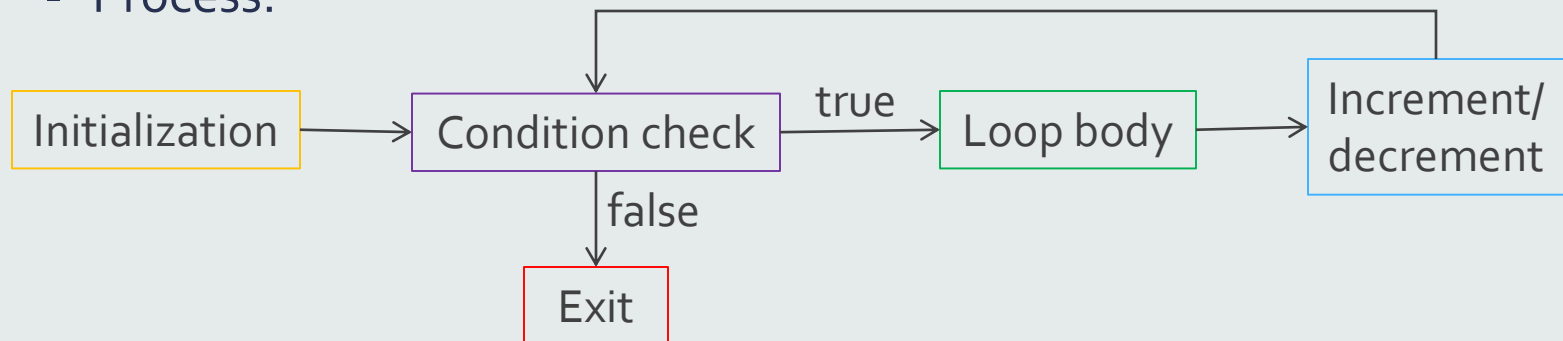
for each or Enhanced for loop

- Syntax
  for(datatype var1: var2){
          Loop body
  }

- var2 should be either an array or collection. var1 should be a type of var2.

- This is most convenient loop to retrieve element of Arrays and Collections.

break, continue and return

- break is used to come out of a loop or switch.

- continue is used to skip the rest of current iteration and go to condition check of next iteration in loops.

- return is used to return value to its calling method.

- A method can return only one value.