Understanding for-each loop

- For each Introduced in 1.5version, It acts as an alternative to for loop, while loop etc for retrieving the elements in the array.
- By using for-each loop we can easily retrieve the elements easily from the arrays.
- It cannot traverse the elements in reverse order because it does not work on index values
- For-each also acts as alternative for iterator when retrieving elements from collections.

Syntax:

- Step 1 :-Declare a variable that is the same type as the base type of the array
- Step 2 :-Write the Colon (:)
- Step 3 :-Then write the array name
- Step 4 :-In the loop body we have to use the variable which we have created

Jump Statements

- Java jump statements enable transfer of control to other parts of program.
- Java provides three jump statements:
  1) break
  2) continue
  3) return
- In addition, Java supports exception handling that can also alter the control flow of a program.

Understanding break statement

- The break statement is used to jump out of a loop.
- When the break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
- We can us break statement inside "switch", "loops" & "labeled blocks"., other than this if you are using any where you will be getting an compile time error.

Understanding continue statement

- Continue statement is used to skip current iteration and continue for the next iteration in the loop.
- We can us continue statement inside "loops" & "labeled blocks"., other than this if you are using any where you will be getting an compile time error.