Topic Covered:
The break and continue keywords, The break and continue keywords with label, The foreach Loop and Local Variable Type Inference in for Loop

THE BREAK KEYWORD

The break statement in Java programming language has the following two usages:

- 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.
- It can be used to terminate a case in the switch statement.
- It can also be used with a label like goto statement is C language

THE CONTINUE KEYWORD

- The continue keyword can be used in any of the loop control structures. It causes the loop to immediately jump to the next iteration of the loop.
- In a for loop, the continue keyword causes control to immediately jump to the update statement.
- In a while loop or do/while loop, control immediately jumps to the Boolean expression.
- The continue may specify a label to describe which enclosing loop to continue.

PROGRAM 1: Using break to exit a for loop.

```
class BreakLoop
{
public static void main(String args[])
{
for(int i=0; i<100; i++) {
   if(i == 10) break; // terminate loop if i is 10
   System.out.println("i: " + i);
   }
System.out.println("Loop complete.");
}
}
```

PROGRAM 2: Using break to exit a while loop.

```
class BreakLoop2
{
public static void main(String args[])
{
int i = 0;
while(i < 100) {
   if(i == 10) break; // terminate loop if i is 10
   System.out.println("i: " + i);
   i++;
   }
System.out.println("Loop complete.");
}
```

```
}
```

PROGRAM 3: Test for primes.

```
class FindPrime
{
public static void main(String args[])
{
int num = 14;
boolean isPrime = true;
for(int i=2; i <= num/i; i++) {
   if((num % i) == 0) {
      isPrime = false;
      break;
      }
   }
if(isPrime)
   System.out.println("Prime");
else
   System.out.println("Not Prime");
}
}
```

PROGRAM 4: Using break with nested loops.

```
class BreakLoop3
{
public static void main(String args[])
{
for(int i=0; i<3; i++) {
   System.out.print("Pass " + i + ": ");
   for(int j=0; j<100; j++) {
      if(j == 10) break; // terminate loop if j is 10
      System.out.print(j + " ");
      }
   System.out.println();
   }
System.out.println("Loops complete.");
}
}
```

PROGRAM 5: Using break as a civilized form of goto.

```
class Break
{
public static void main(String args[])
{
boolean t = true;
first: {
   second: {
      third: {
         System.out.println("Before the break.");
         if(t) break second; // break out of second block
         System.out.println("This won't execute");
         }
      System.out.println("This won't execute");
      }
   System.out.println("This is after second block.");
   }
}
}
```

PROGRAM 6: Using break to exit from nested loops

```
class BreakLoop4
{
public static void main(String args[])
{
outer: for(int i=0; i<3; i++) {
         System.out.print("Pass " + i + ": ");
            for(int j=0; j<100; j++) {
               if(j == 10) break outer; // exit both loops
               System.out.print(j + " ");
            }
         System.out.println("This will not print");
         }
System.out.println("Loops complete.");
}
}
```

PROGRAM 7: Demonstrate continue.

```
class Continue {
public static void main(String args[]) {
for(int i=0; i<10; i++) {
   System.out.print(i + " ");
   if (i%2 == 0) continue;
   System.out.println("");
   }
}
}
```

PROGRAM 8: Using continue with a label.

```
class ContinueLabel
{
public static void main(String args[])
{
outer: for (int i=0; i<10; i++) {
          for(int j=0; j<10; j++) {
             if(j > i) {
                System.out.println();
                continue outer;
             }
             System.out.print(" " + (i * j));
          }
       }
System.out.println();
}
}
```

THE FOREACH LOOP

JDK 1.5 introduced a new for loop known as foreach loop or enhanced for loop, which enables you to traverse the complete array sequentially without using an index variable. Syntax:

```
for(declaration : expression) {
// Statements
}
```

Declaration: A newly declared block variable, is of a type compatible with the elements of the array you are accessing.
Expression: The expression can be an array variable or method call that returns an array.

PROGRAM 9: Foreach Loop for array

```
class TestArray
{
  public static void main(String[] args)
{
  double[] myList = {1.9, 2.9, 3.4, 3.5};
  // Print all the array elements
  for (double element: myList) {
     System.out.println(element);
    }
  }
}
```

PROGRAM 10: Displaying 2D array with foreach loop

```
class TestArray{
public static void main(String args[]){

//declaring and initializing 2D array
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};

//printing 2D array
for(int a[] : arr){
   for(int b: a ){
      System.out.print(b);
      }
   System.out.println();
   }
}
}
```

LOCAL VARIABLE TYPE INFERENCE IN A FOR LOOP

Local variable type inference can be used in a for loop when declaring and initializing the loop control variable inside a traditional for loop, or when specifying the iteration variable in a for-each for.

PROGRAM 11: Use type inference in for loop

```
class TypeInf{
public static void main(String args[]){

System.out.println("Values of x: ");
for(var x = 2.5; x < 100.0 ; x *= 2)
   System.out.print(x + ", ");

System.out.println();

int arr[]= { 1, 2, 3, 4, 5 };
System.out.println("Values in Array: ");
for(var v : arr)
   System.out.print(v + ", ");

}
}
```

EXERSICE 6-1:

Create a program for Mark sheet with following requirements:

- Take input for data of 5 students in 7 different arrays like: (Roll Number, Name, Marks of 5 different subjects)
- Calculate and store the Total, Percentage and Grade of them in other 3 arrays.
- Use loops, and decision statements for the repetition and decision making.
- Then display the results as shown below

```
Roll No.:        2k19/SW/130
Name of Student: XYZ
English:         40          Calculus:        50
Electronics:     60          Programming:     70
Intro. To IT:    80
Total:           300         Percentage:      60%         Grade:     B

Roll No.:        2K19/SW/131
N...
......
```

EXERCISE 6-2:

Write a program to input a character from user and check whether given character alphabet, digit or special character is using if else. If user presses ENTER the program should exit.

```
Input character: a
'a' is alphabet
Input character: A
'A' is alphabet
Input character: #
'#' is special character
Input character: 1
'1' is a number
Input character:
Exit
```