The for-loop and Nested loops
Outline

- The for Statement Syntax
- Semantics of the for Statement
- Nested Loops
- continue, break, and exit Statements
The for Statement Syntax

for (Initializing; Boolean_Expression; Update)
Block

- Note that the three control expressions are separated by two, not three, semicolons
- Note that there is no semicolon after the closing parenthesis at the beginning of the loop
The **for** statement is most commonly used to step through an integer variable in equal increments.

It begins with the keyword **for**, followed by three expressions in parentheses that describe what to do with one or more *controlling variables*.

- The first expression tells how the control variable or variables are *initialized* or *declared* and *initialized* before the first iteration.
- The second expression determines when the loop should *end*, based on the evaluation of a Boolean expression *before* each iteration.
- The third expression tells how the control variable or variables are *updated after* each iteration of the loop body.
Nested Loops

Loops can be nested, just like other Java structures.
- When nested, the inner loop iterates from beginning to end for each single iteration of the outer loop.

```
for (Initializing; Boolean_Expression; Update)
  Block 1
```

- Block 1 can contain other loop statements as follows.

```
Block 1 ➔ for (Initializing; Boolean_Expression; Update)
  Block 2
```
Loops can be *nested*, just like other Java structures

When nested, the inner loop iterates from beginning to end for each single iteration of the outer loop

```java
int rowNum, columnNum;
for (rowNum = 1; rowNum <=3; rowNum++)
{
    for (columnNum = 1; columnNum <=2; columnNum++)
    {
        System.out.print(" row " + rowNum + " column " + columnNum);
        System.out.println();
    }
}
```
- continue, break, and exit Statements …

```java
Class test {
    public static void main( String [] args) {
        for (int I = 0; I < 10; i++) {
            statement 1;
            statement 2;
            if (cond) continue;
            statement 3;
            statement 4;
        }
        statement 5;
        statement 6;
    }
}
```
Class test {
    public static void main(String[] args) {
        for (int I = 0; I < 10; i++) {
            statement 1;
            statement 2;
            if (cond) break;
            statement 3;
            statement 4;
        }
        statement 5;
        statement 6;
    }
}
\[ \text{Class test} \{ \\
\hspace{1em} \text{public static void main(String [] args) \{ \\
\hspace{2em} \text{for (int I = 0; I < 10; i++) \{ \\
\hspace{3em} \text{statement 1;} \\
\hspace{3em} \text{statement 2;} \\
\hspace{3em} \text{if( cond) exit;} \\
\hspace{3em} \text{statement 3;} \\
\hspace{3em} \text{statement 4;} \\
\hspace{2em} \}} \\
\hspace{1em} \text{statement 5;} \\
\hspace{1em} \text{statement 6;} \\
\hspace{1em} \} \\
\} \]
For-loop examples
Questions

1. Write a Java program which computes the sum of all the odd numbers between 0 and 100.

2. Write a Java program which reads 20 numbers using a scanner and computes their average.

3. Write a Java program which reads unknown number of integers using a scanner and counts the number of odd numbers and the number of even numbers. Assume the input integers are all positive. Use a negative number as a sentinel.
Q1 Solution

Write a Java program which computes the sum of all the odd numbers between 0 and 100.

```java
int sum = 0;
for (int n = 1; n <= 100; n = n + 2) {
    sum += n;
}
System.out.println("The sum is \( + sum");
```
Q2 Solution

Write a Java program which reads 20 numbers using a scanner and computes their average.

```java
Scanner kb = new Scanner(System.in);
double x;
double sum = 0;
While (int cnt = 0; cnt < 20; cnt++) {
    System.out.println("Enter a number");
    x = kb.nextDouble();
    sum += x;
}
System.out.println("The Average is " + sum/cnt);
```
Q3 Solution

Write a Java program which reads unknown number of integers using a scanner and counts the number of odd numbers and the count of even numbers. Assume the input integers are all positive. Use any negative number as a sentinel.

```java
Scanner kb = new Scanner(System.in);
int even_cnt = 0;
int odd_cnt = 0;
int n;
For(;;) {
    n = kb.nextInt();
    if (n < 0)
        break;
    else if ( mod(n,2) == 0)
        even_cnt++;
    else
        odd_cnt++;
}
System.out.println("Even = " + even_cnt + " odd = " odd_cnt);
```
Nested-loop examples
Questions

1. Write a java program which gives the following output
   1
   22
   333
   4444
   55555

2. Write a java program which prints all the prime numbers less than 1000.
Write a java program which gives the following output

1
22
333
4444

```java
for(int k = 1; k <= 5; k++) {
    for (int j = 1; j <=k; j++)
        System.out.print(k);
    System.out.println();
}
```
Write a java program which prints all the prime numbers less than 1000.

```java
int n, j;
for(int k = 2; k < 100; k++) {
    n = 0;
    j = 2;
    while(n == 0 && j < k/2) {
        if (mod(k, j) == 0) n++;
        j++;
    }
    if( n ==0) System.out.println(k);
}
```
THE END