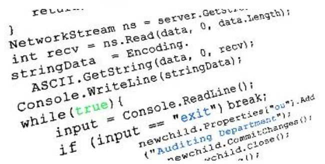
برمجة I
Programming I

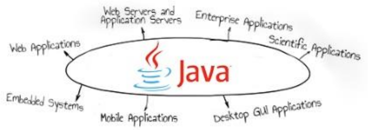
Ahmad Khoj
King Abdulaziz University

KING ABDULAZIZ UNIVERSITY

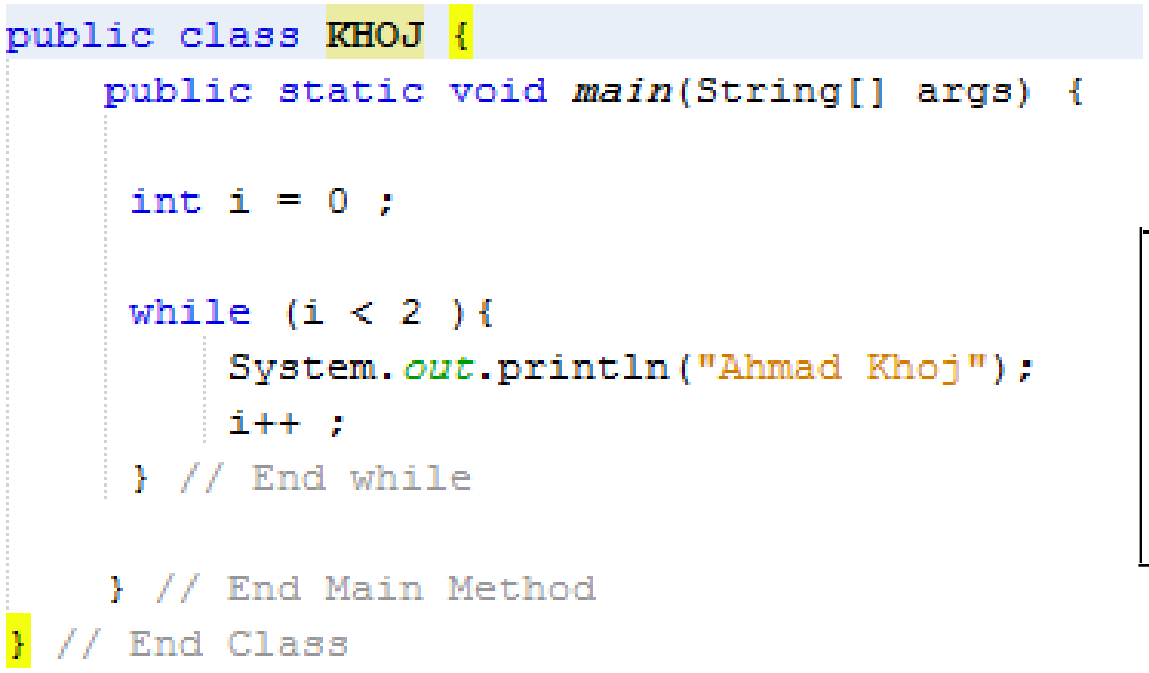
Chapter 4 Loops



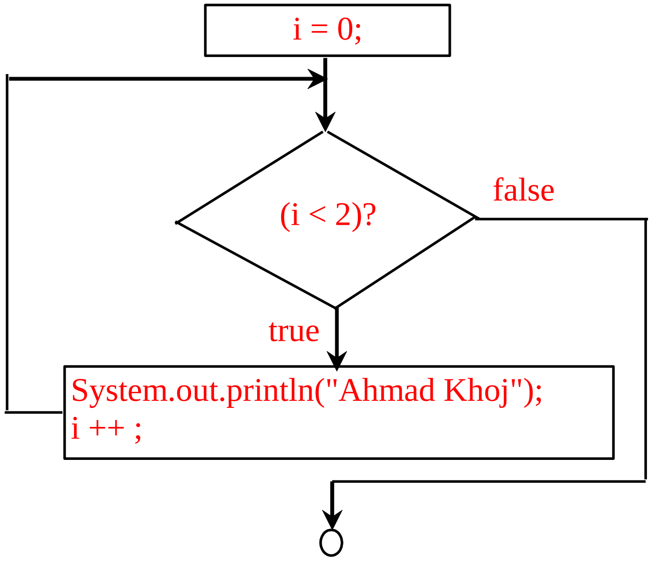
```
}
NetworkStream ns = server.GetStr
int recv = ns.Read(data, 0, data.Length);
stringData  = Encoding.
  ASCII.GetString(data, 0, recv);
Console.WriteLine(stringData);
while(true){
    input = Console.ReadLine();
    if (input == "exit") break;
        newchild.Properties["ou"].Add
        ("Auditing Department");
        newchild.CommitChanges();
        newchild.Close();
```



while Loop Flow Chart



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

      int i = 0 ;

      while (i < 2 ){
          System.out.println("Ahmad Khoj");
          i++ ;
      } // End while

    } // End Main Method
} // End Class
```



```
Ahmad Khoj
Ahmad Khoj
```

animation

Trace while Loop

Initialize i

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

animation

Trace while Loop, cont.

(i < 2) is true

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

Trace while Loop, cont.

Print Ahmad Khoj

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

Trace while Loop, cont.

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

Increase i by 1
i is 1 now

Trace while Loop, cont.

(i < 2) is still true since i is 1

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

animation

Trace while Loop, cont.

Print Ahmad Khoj

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

Trace while Loop, cont.

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

Increase i by 1
i is 2 now

Trace while Loop, cont.

(i < 2) is false since i is 2 now

```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

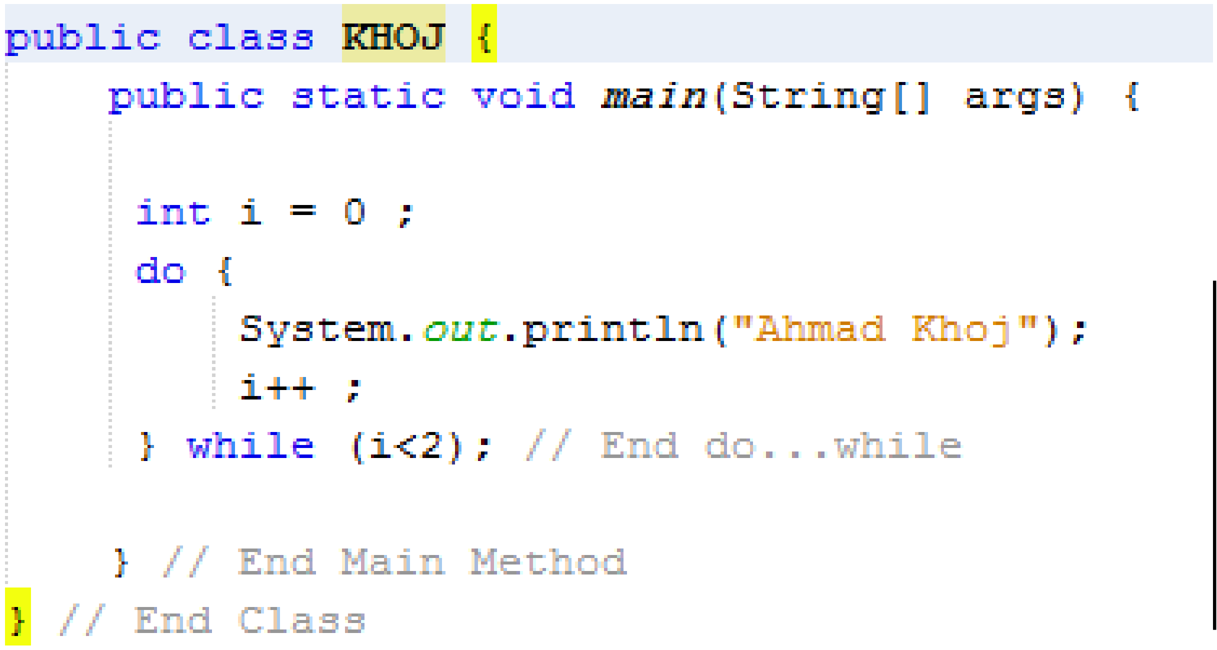
animation

Trace while Loop

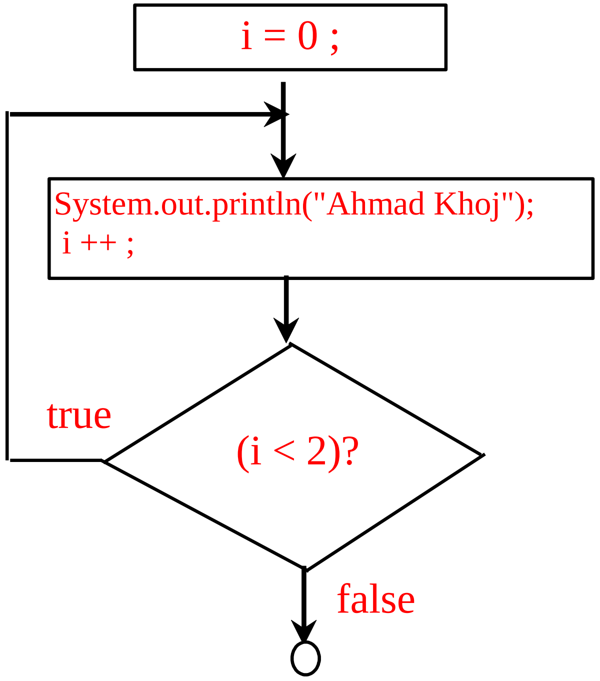
```
int i = 0;
while (i < 2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

The loop exits. Execute the next statement after the loop.

`do-while` Loop

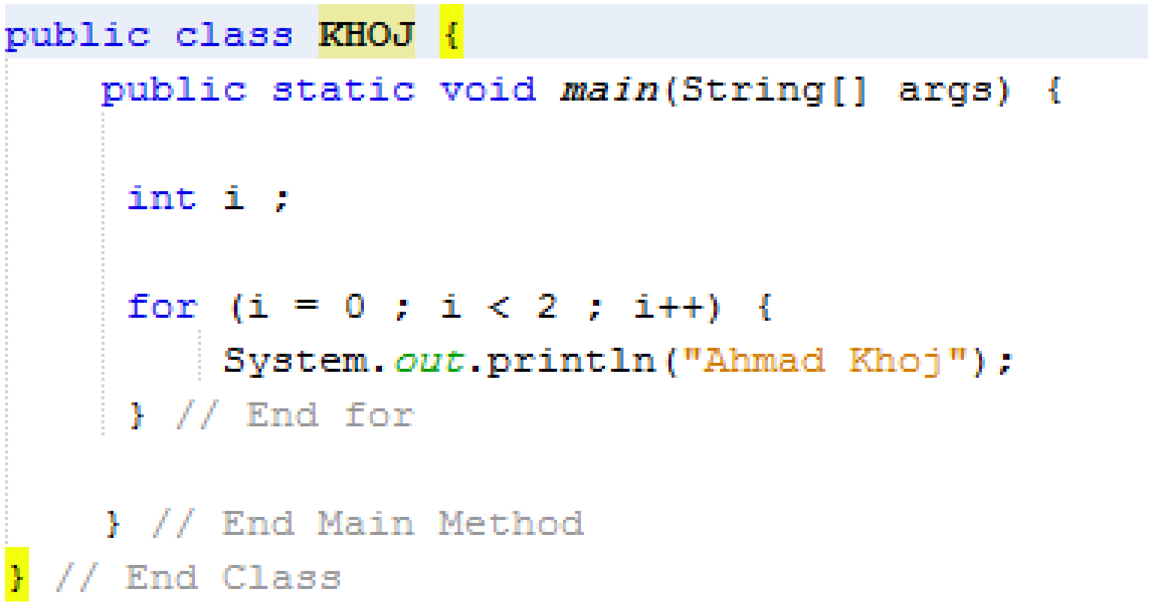


```
public class KHOJ {
    public static void main(String[] args) {

        int i = 0 ;
        do {
            System.out.println("Ahmad Khoj");
            i++ ;
        } while (i<2); // End do...while

    } // End Main Method
} // End Class
```



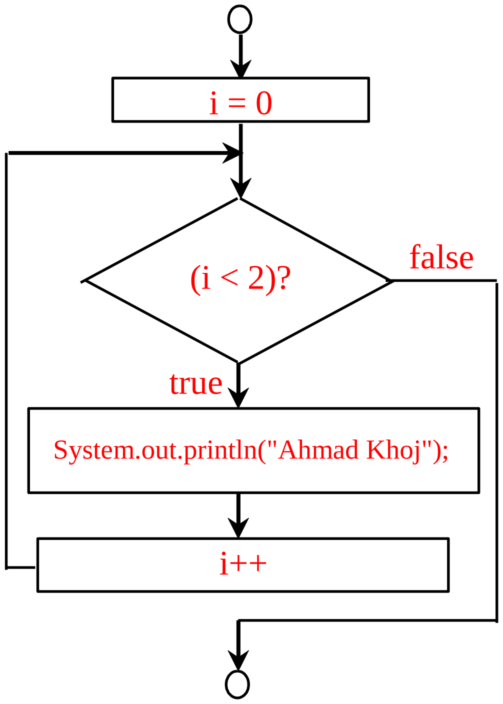
```
Ahmad Khoj
Ahmad Khoj
```

for Loops



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

     int i ;

     for (i = 0 ; i < 2 ; i++) {
         System.out.println("Ahmad Khoj");
     } // End for

    } // End Main Method
} // End Class
```

```
Ahmad Khoj
Ahmad Khoj
```



Trace for Loop

Declare i

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

Trace for Loop, cont.

Execute initializer
i is now 0

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

animation

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

(i < 2) is true
since i is 0

animation

Trace for Loop, cont.

Print Ahmad Khoj

```
int i;
for (i = 0; i < 2; i++) {
 System.out.println("Ahmad Khoj");
}
```

Trace for Loop, cont.

Execute adjustment statement
i now is 1

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

Trace for Loop, cont.

(i < 2) is still true
since i is 1

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

animation

Trace for Loop, cont.

Print Ahmad Khoj

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

animation

Trace for Loop, cont.

Execute adjustment statement
i now is 2

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

animation

Trace for Loop, cont.

(i < 2) is false
since i is 2

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

animation

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Ahmad Khoj");
}
```

Exit the loop. Execute the next statement after the loop

Which Loop to Use?

- The three forms of loop statements, `while`, `do-while`, and `for`, are expressively equivalent; that is, you can write a loop in any of these three forms.
  - For example, a while loop in (a) in the following figure can always be converted into the following for loop in (b):

```
while (i<2) {
  System.out.println("Ahmad Khoj");
}
```

(a)

Equivalent

```
for ( ; i<2 ; ) {
  System.out.println("Ahmad Khoj");
}
```

(b)

- A for loop in (a) in the following figure can generally be converted into the following while loop in (b) except in certain special cases:

```
for (int i=0 ; i<2 ; i++) {
  System.out.println("Ahmad Khoj");
}
```

(a)

Equivalent

```
int i=0;
while (i<2) {
  System.out.println("Ahmad Khoj");
  i++;
}
```

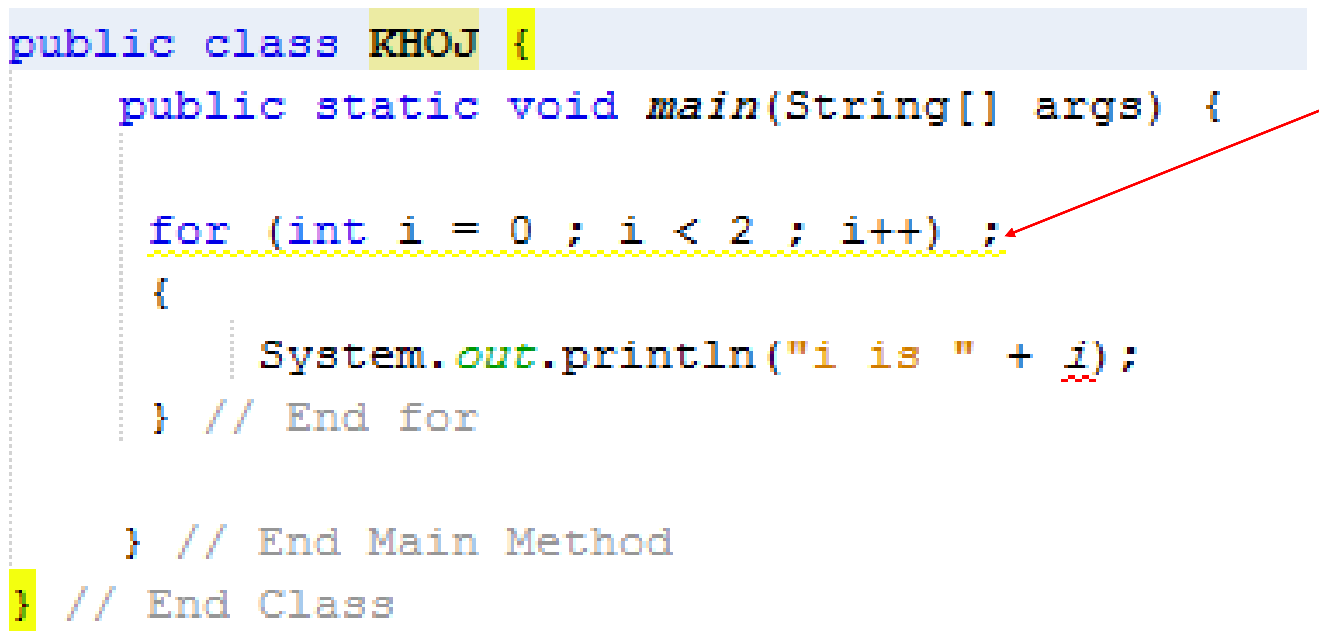
(b)

Recommendations

- Use the one that is most intuitive and comfortable for you. In general:
  - A `for` loop may be used if the number of repetitions is known, as, for example, when you need to print a message 100 times.
  - A `while` loop may be used if the number of repetitions is not known, as in the case of reading the numbers until the input is 0.
  - A `do-while` loop can be used to replace a while loop if the loop body has to be executed before testing the continuation condition.

Caution

Adding a semicolon at the end of the `for` clause before the loop body is a common mistake, as shown below:

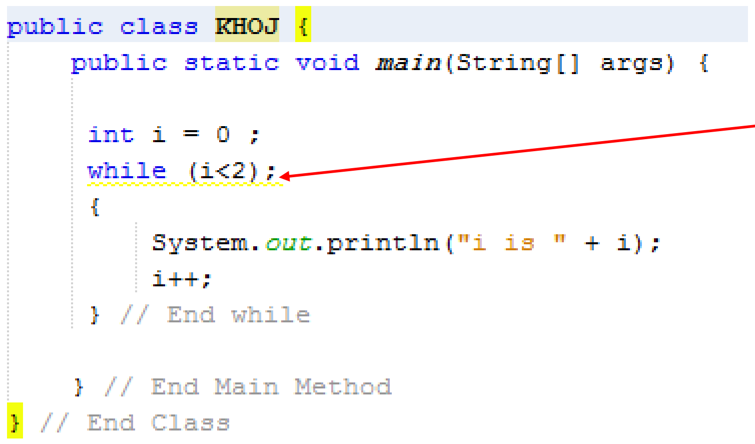


```
public class KHOJ {
    public static void main(String[] args) {

        for (int i = 0 ; i < 2 ; i++) ;
        {
            System.out.println("i is " + i);
        } // End for

    } // End Main Method
} // End Class
```

Logic Error

Caution, cont.

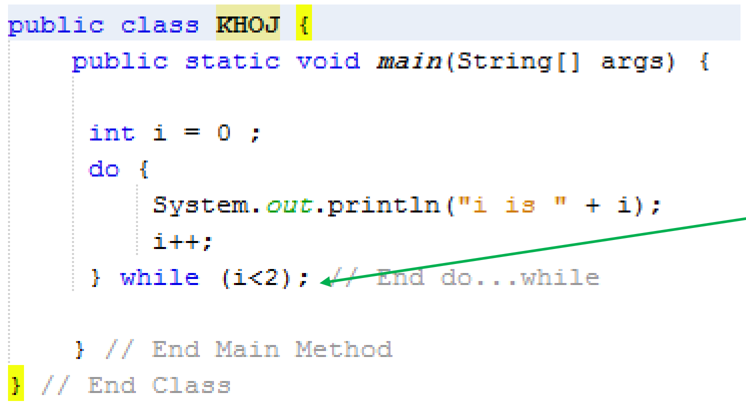
Similarly, the following loop is also wrong:



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

     int i = 0 ;
     while (i<2);
     {
         System.out.println("i is " + i);
         i++;
     } // End while

    } // End Main Method
} // End Class
```

Logic Error

In the case of the do-while loop, the following semicolon is needed to end the loop.



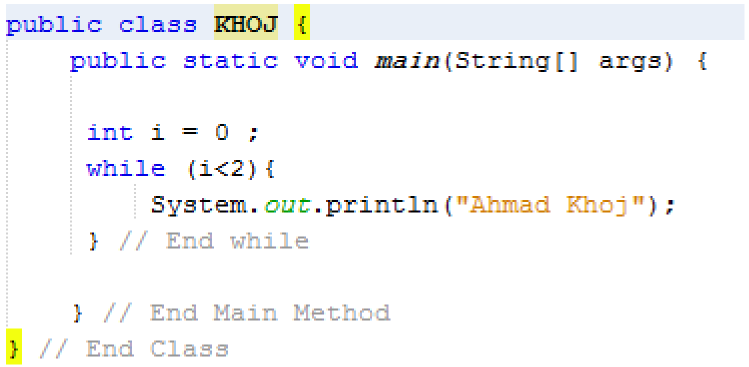
```
public class KHOJ {
    public static void main(String[] args) {

     int i = 0 ;
     do {
         System.out.println("i is " + i);
         i++;
     } while (i<2); // End do...while

    } // End Main Method
} // End Class
```

Correct

Infinite loop

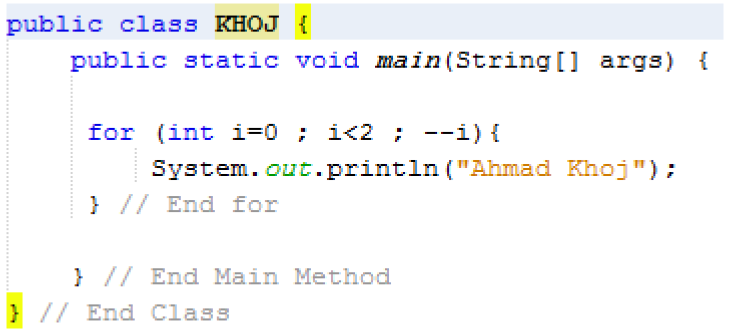
Infinite loop is a loop statement that executes infinitely.

1. while loop:



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

     int i = 0 ;
     while (i<2){
         System.out.println("Ahmad Khoj");
     } // End while

    } // End Main Method
} // End Class
```

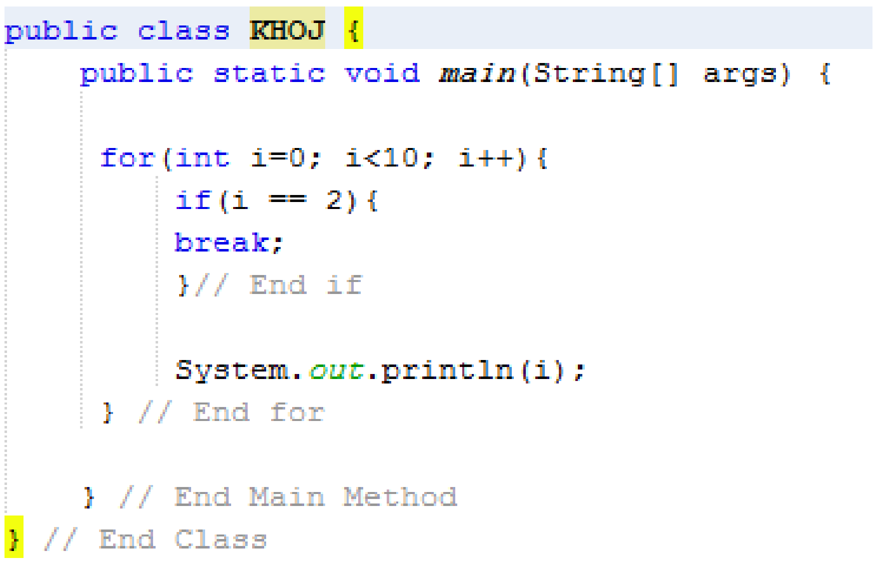
2. for loop:



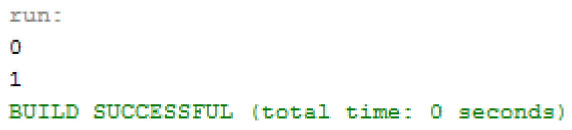
```
public class KHOJ {
    public static void main(String[] args) {

     for (int i=0 ; i<2 ; --i){
         System.out.println("Ahmad Khoj");
     } // End for

    } // End Main Method
} // End Class
```

Using break and continue

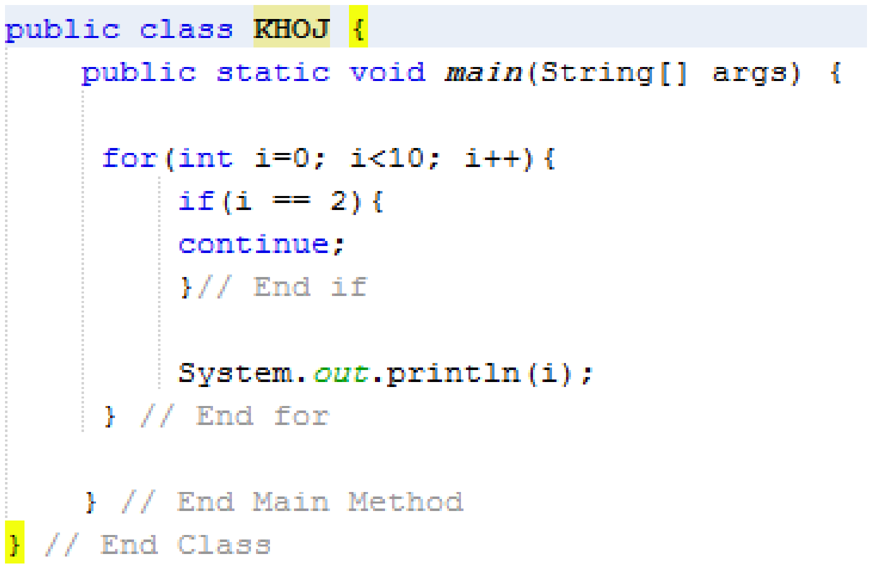
Examples for using the break and continue keywords:



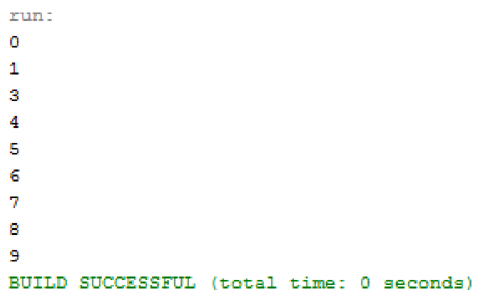
```
public class KHOJ {
    public static void main(String[] args) {

     for(int i=0; i<10; i++){
         if(i == 2){
         break;
         }// End if

         System.out.println(i);
     } // End for

    } // End Main Method
} // End Class
```



```
run:
0
1
BUILD SUCCESSFUL (total time: 0 seconds)
```

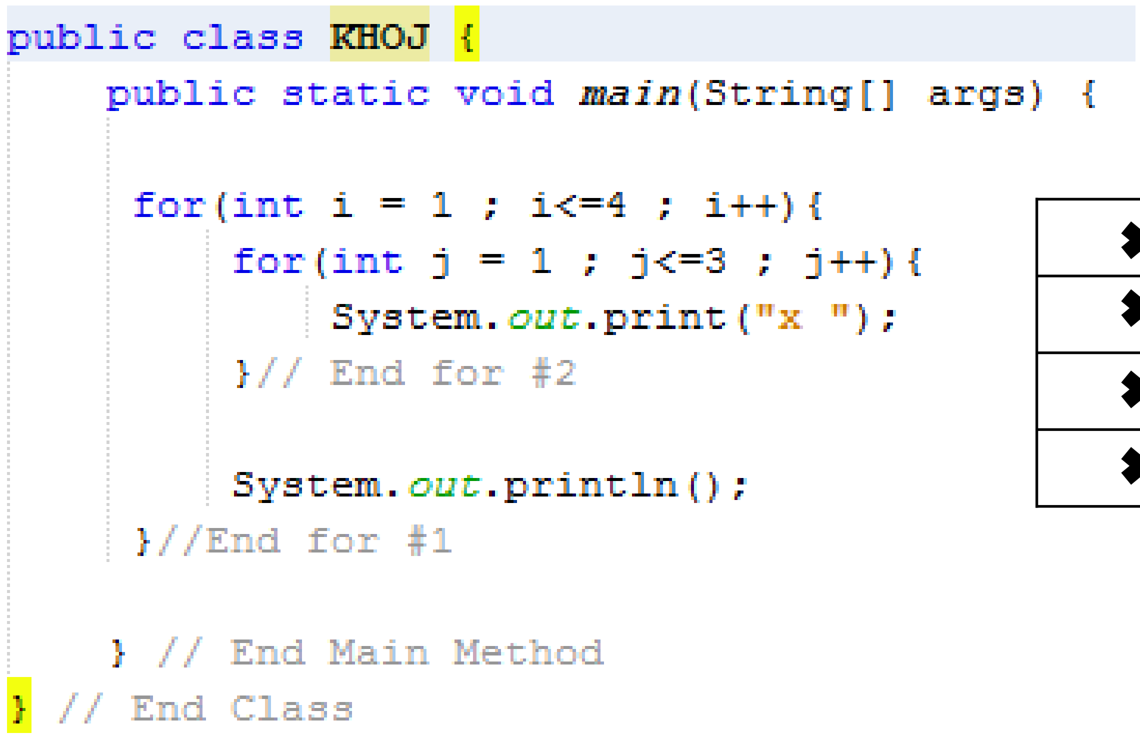


```
public class KHOJ {
    public static void main(String[] args) {

     for(int i=0; i<10; i++){
         if(i == 2){
         continue;
         }// End if

         System.out.println(i);
     } // End for

    } // End Main Method
} // End Class
```



```
run:
0
1
3
4
5
6
7
8
9
BUILD SUCCESSFUL (total time: 0 seconds)
```

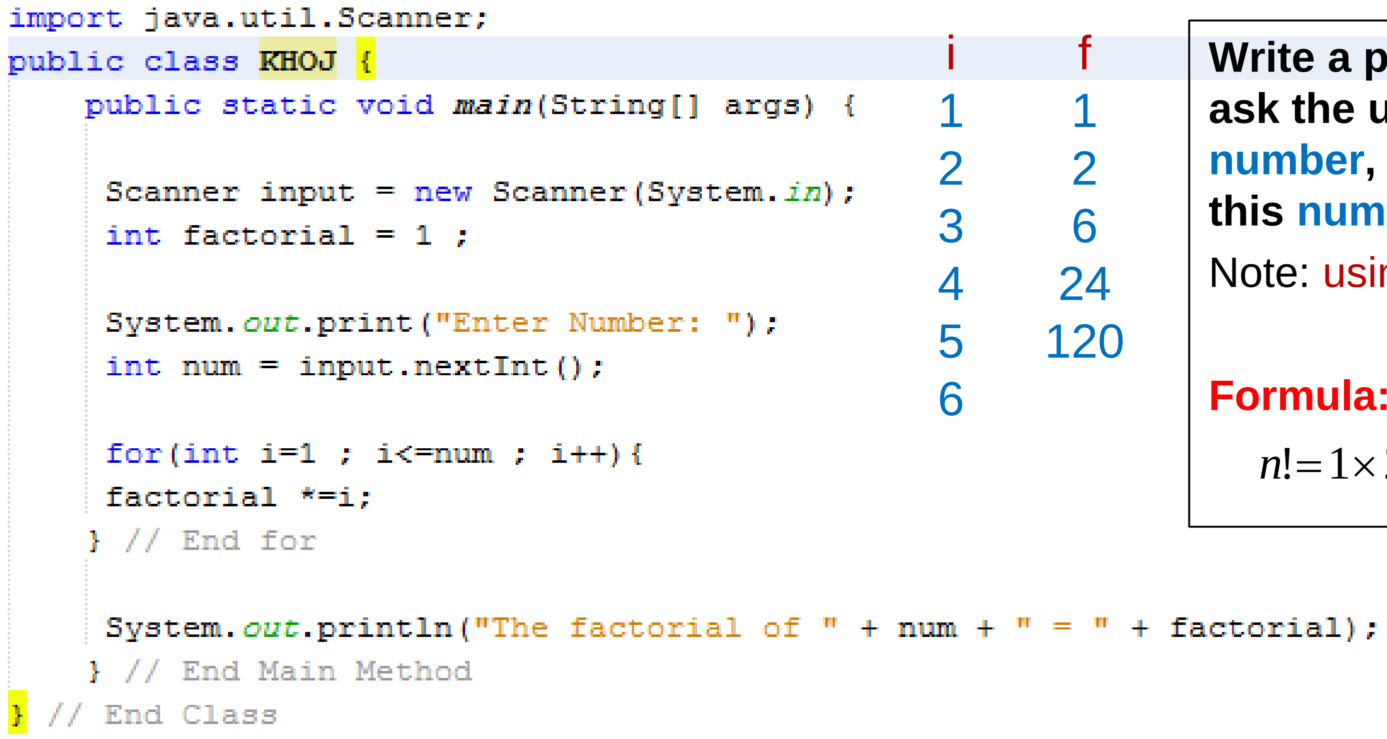
Nested Loops



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

     for(int i = 1 ; i<=4 ; i++){
         for(int j = 1 ; j<=3 ; j++){
             System.out.print("x ");
         }// End for #2

         System.out.println();
     }//End for #1

    } // End Main Method
} // End Class
```

✖	✖	✖
✖	✖	✖
✖	✖	✖
✖	✖	✖

Factorial



```
import java.util.Scanner;
public class KHOJ {
    public static void main(String[] args) {

     Scanner input = new Scanner(System.in);
     int factorial = 1 ;

     System.out.print("Enter Number:  ");
     int num = input.nextInt();

     for(int i=1 ; i<=num ; i++){
     factorial *=i;
    } // End for

     System.out.println("The factorial of " + num + " = " + factorial);
    } // End Main Method
} // End Class
```

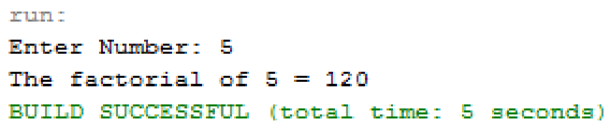
i	f
1	1
2	2
3	6
4	24
5	120
6	

Write a program in Java to ask the user to enter a number, so the factorial of this number will be displayed

Note: using the Scanner class

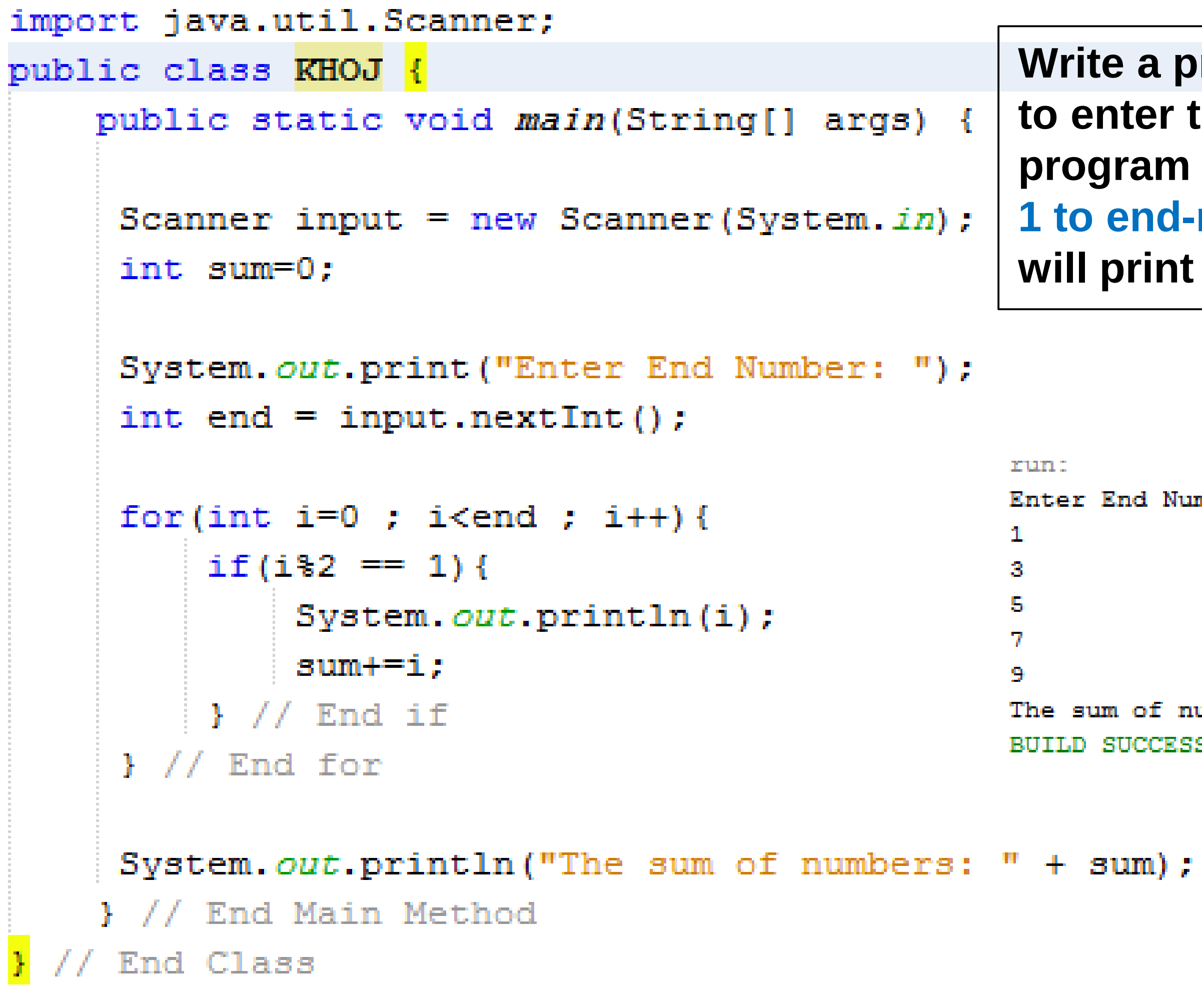
Formula:

$$n! = 1 \times 2 \times 3 \times 4 \times \ldots \times n$$



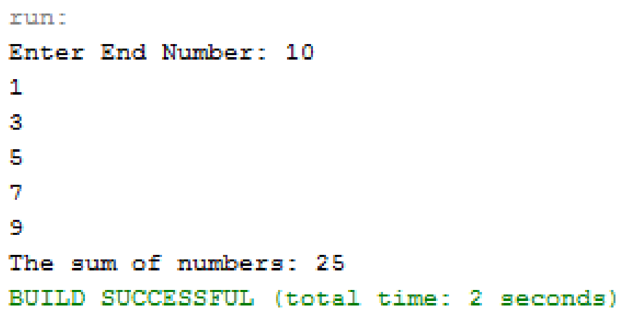
```
run:
Enter Number: 5
The factorial of 5 = 120
BUILD SUCCESSFUL (total time: 5 seconds)
```

Odd Numbers



```
import java.util.Scanner;
public class KHOJ {
    public static void main(String[] args) {

     Scanner input = new Scanner(System.in);
     int sum=0;

     System.out.print("Enter End Number: ");
     int end = input.nextInt();

     for(int i=0 ; i<end ; i++){
         if(i%2 == 1){
             System.out.println(i);
             sum+=i;
         } // End if
     } // End for

     System.out.println("The sum of numbers: " + sum);
    } // End Main Method
} // End Class
```

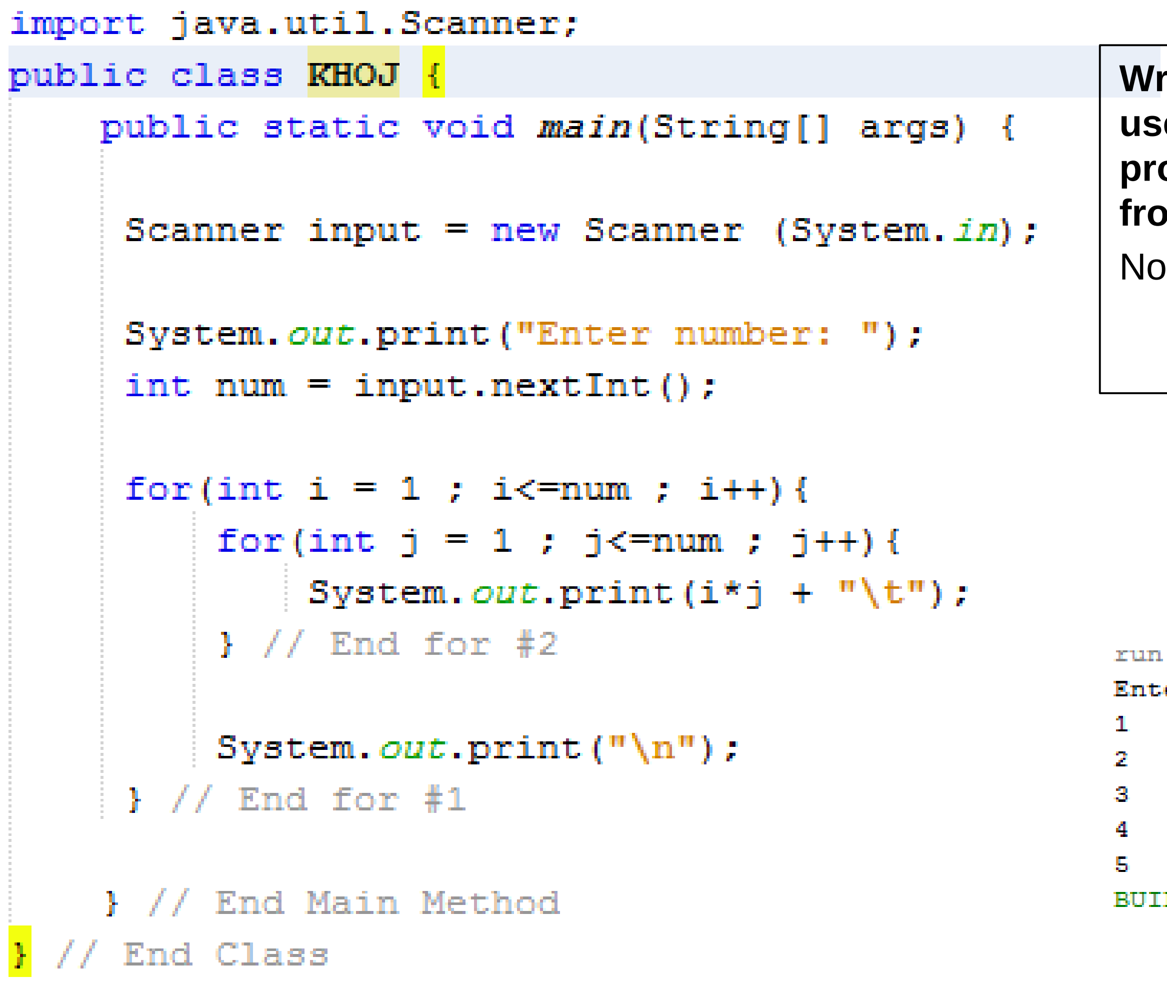
Write a program in Java to ask the user to enter the end-number, and then the program will display odd numbers from 1 to end-number. Finally, the program will print the sum of these numbers.



```
run:
Enter End Number: 10
1
3
5
7
9
The sum of numbers: 25
BUILD SUCCESSFUL (total time: 2 seconds)
```

i	S
0	
1	1
2	
3	4
4	
5	9
6	
7	16
8	
9	25
10	

Multiplication Table

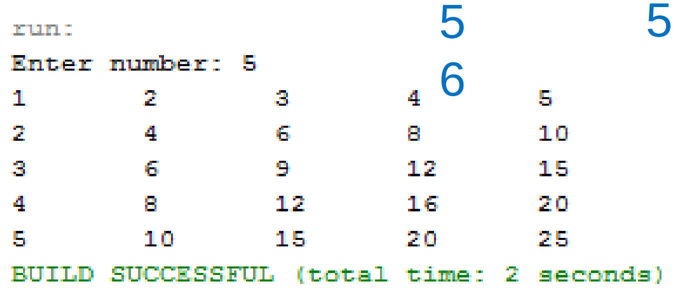


```java
import java.util.Scanner;
public class KHOJ {
    public static void main(String[] args) {

     Scanner input = new Scanner (System.in);

     System.out.print("Enter number: ");
     int num = input.nextInt();

     for(int i = 1 ; i<=num ; i++){
         for(int j = 1 ; j<=num ; j++){
             System.out.print(i*j + "\t");
         } // End for #2

         System.out.print("\n");
     } // End for #1

    } // End Main Method
} // End Class
```

Write a program in Java to ask the user to enter a number, and then the program to print multiplication table from 1 to the entered number.

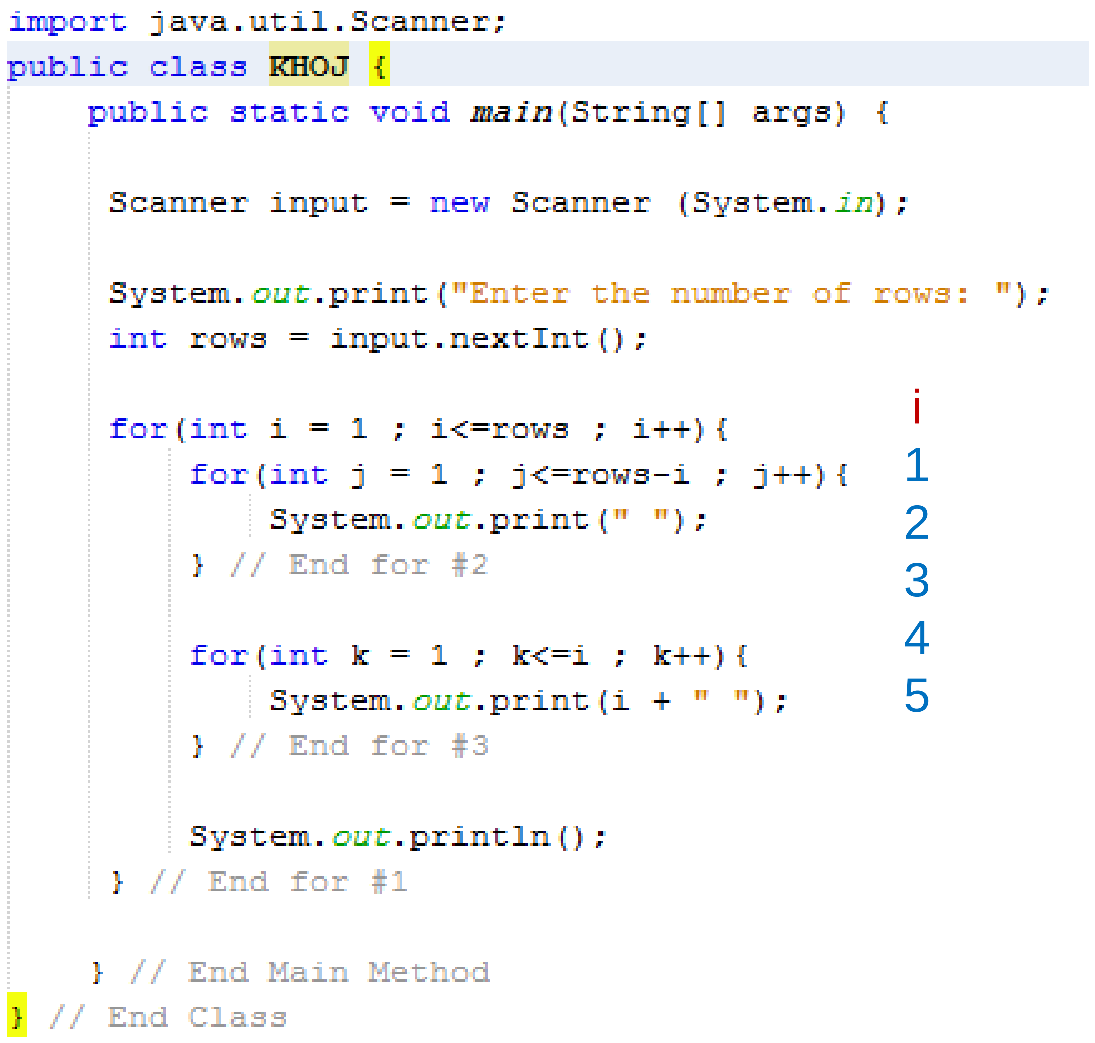
Note: using the Scanner class

```
i    j
1    1
2    2
3    3
4    4
5    5
6
```



```
run:
Enter number: 5
1       2       3       4       5
2       4       6       8       10
3       6       9       12      15
4       8       12      16      20
5       10      15      20      25
BUILD SUCCESSFUL (total time: 2 seconds)
```

Pyramid of number

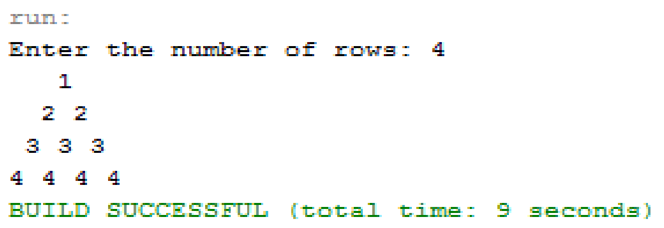


```
import java.util.Scanner;
public class KHOJ {
    public static void main(String[] args) {

     Scanner input = new Scanner (System.in);

     System.out.print("Enter the number of rows: ");
     int rows = input.nextInt();

     for(int i = 1 ; i<=rows ; i++){
         for(int j = 1 ; j<=rows-i ; j++){
             System.out.print(" ");
         } // End for #2

         for(int k = 1 ; k<=i ; k++){
             System.out.print(i + " ");
         } // End for #3

         System.out.println();
     } // End for #1

    } // End Main Method
} // End Class
```

Write a program in Java to make like the following pattern with a number repeated in the same row. The number of rows should be entered by the user

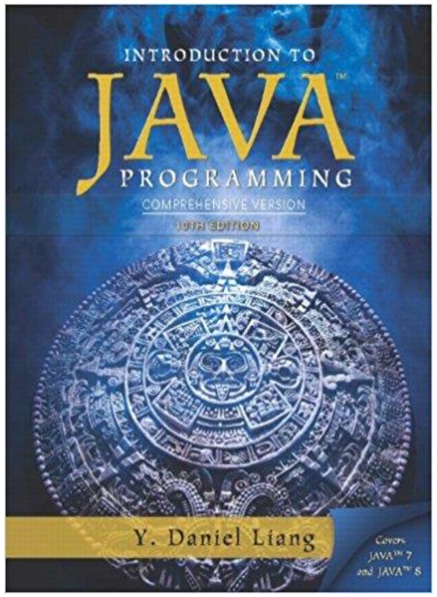
Note: using the Scanner class

i	j	k
1	1	1
2	2	2
3	3	3
4		4
5		



```
run:
Enter the number of rows: 4
   1
  2 2
 3 3 3
4 4 4 4
BUILD SUCCESSFUL (total time: 9 seconds)
```

Introduction to Java Programming
(10th Edition)



By
Y. Daniel Liang