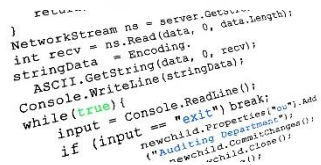
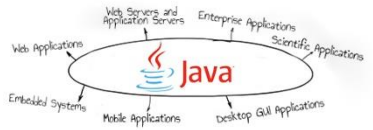
برمجة I

Programming I

Ahmad Khoj
King Abdulaziz University

Chapter 5
Methods



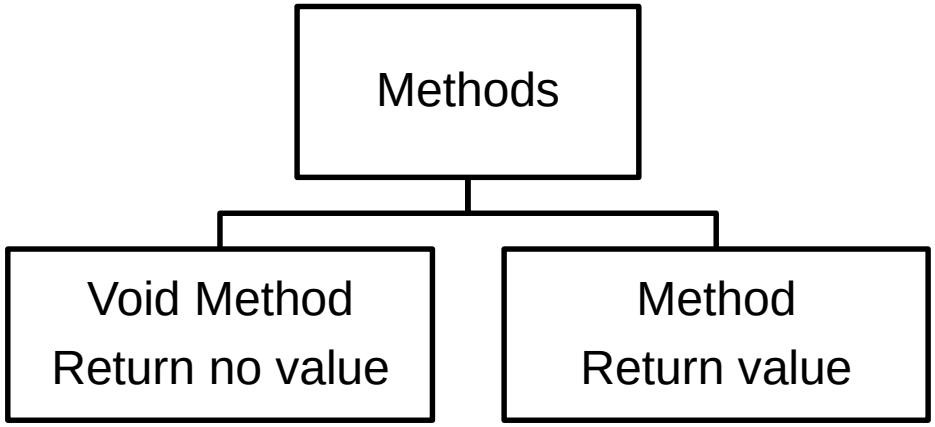


Introducing Methods

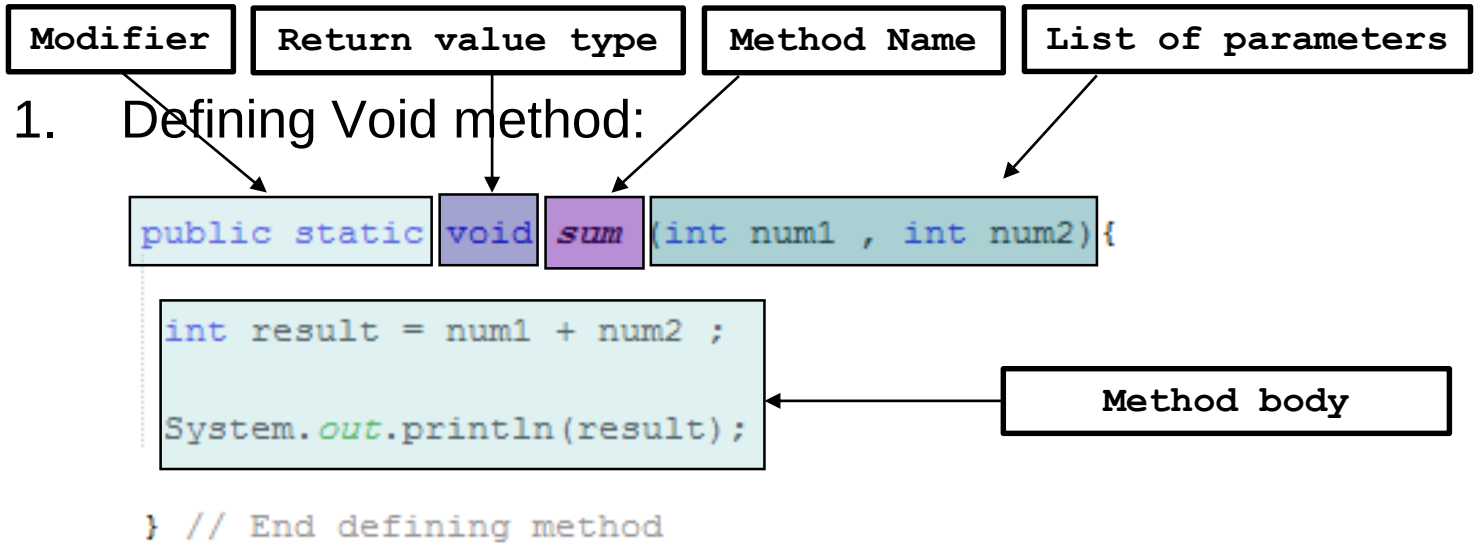
A method is a collection of statements that are grouped together to perform an operation.

- **Benefits of Methods:**

1. Write a method once and reuse it anywhere.
2. Information hiding. Hide the implementation from the user.
3. Reduce complexity.



The syntax for defining a method



1. Defining Void method:

2. Defining method return value:

```
public static int sum (int num1 , int num2){

 int result = num1 + num2 ;

 return result ;

} // End defining method
```

Note:

You have to define a method:

1. Inside a Class
2. Outside a main method

Defining Void

```
public class KHOJ {

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
         sum(num1 , num2);
         System.out.println("Ahmad Khoj");
    } // End Main Method

    public static void sum (int num1 , int num2){
     int result = num1 + num2 ;
     System.out.println(result);
    } // End sum Method

} // End Class
```

2) Calling Method

1) Defining Method

Output:
7
Ahmad Khoj

Defining Void

```
public class KHOJ {

    public static void sum (int num1 , int num2){
     int result = num1 + num2 ;
     System.out.println(result);
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
         sum(num1 , num2);
         System.out.println("Ahmad Khoj");
    } // End Main Method

} // End Class
```

1) Defining Method

2) Calling Method

Output:
7
Ahmad Khoj

Defining method return value

```
public class KHOJ {

    public static int sum (int num1 , int num2){
     int result = num1 + num2 ;
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = sum(num1 , num2);
     System.out.println(num3);
     System.out.println("Ahmad Khoj");
    } // End Main Method

} // End Class
```

1) Defining Method

2) Calling Method

Output:
7
Ahmad Khoj

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return resul
    } // End sum Metho

    public static vo  main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

num1 is now 3

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum

    public static voi     n(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

num2 is now 4

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(Stri      gs) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

invoke max(num1 , num2)

invoke max(num1 , num2)
Pass the value of num1 to num1
Pass the value of num2 to num2

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

declare variable result

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

```
public class KHO

    public static in          (int num1 , int num2){
     int result ;

     if(num1 > num2)
          result = num1 ;
     else
          result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

(num1 > num2) is false

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1
     else
        result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

result is now 4

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > nu
         result =
     else
         result = nu

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

Print Ahmad

Output:
Ahmad

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result =
     else
         result = num

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public static void main(String[] args) {
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

return result, which is 4

Output:
Ahmad

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum

    public stati
     int num1 = 3;
     int num2 = 4;
     int num3 = max(num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

return max(num1,num2) and assign the return value to num3

Output:
Ahmad

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public stati
     int num1 = 3;
     int num2 = 4;
     int num3 = ma (num1 , num2);
     System.out.println(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

Print num3 which is 4

Output:
Ahmad
4

```
public class KHOJ {

    public static int max (int num1 , int num2){
     int result ;

     if(num1 > num2)
         result = num1 ;
     else
         result = num2 ;

     System.out.println("Ahmad");
     return result ;
    } // End sum Method

    public stati
     int num1 =
     int num2 = 4;
     int num3 = max  1 , num2);
     System.out.pr tln(num3);
     System.out.println("Khoj");
    } // End Main Method

} // End Class
```

Print Ahmad

Output:
Ahmad
4
Khoj

Example: sum 2 numbers method - return no value

```
public static void main(String[] args){
    int num1 = 3;
    int num2 = 4;
        sum(num1,num2);
}
```

```
public static void sum(int num1,int num2){
 int result = num1 + num2 ;
 System.out.println(result);
}
```

Example: sum 2 numbers method - return int value

```
public static void main(String[] args){
    int num1 = 3;
    int num2 = 4;
    int num3 = sum(num1,num2);
    System.out.println(num3);
}
```

```
public static int sum(int n1,int n2){
 int result = n1 + n2;
 return result;
}
```

Exercise: max method – return no value

```
public static void main(String[] args){
    int n1 = 3;
    int n2 = 4;
        max(n1,n2);
}
```

```
public static void max (int n1,int n2){
 if (n1>n2)
  System.out.println(n1);
 else
  System.out.println(n2);
}
```

Exercise: max method – return int value

```
public static void main(String[] args){
    int n1 = 3;
    int n2 = 4;
    int n3 = max(n1,n2);
    System.out.println(n3);
}
```

```
public static int max(int n1,int n2){
 if (n1>n2)
   return n1;
 else
   return n2;
}
```

Method assists to avoid repeating codes

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

        int sum = 0;
        for (int i = 1; i <= 10; i++)
            sum += i;
        System.out.println("Sum from 1 to 10 is " + sum);

        sum = 0;
        for (int i = 12; i <= 22; i++)
            sum += i;
        System.out.println("Sum from 12 to 22 is " + sum);

        sum = 0;
        for (int i = 32; i <= 40; i++)
            sum += i;
        System.out.println("Sum from 32 to 40 is " + sum);

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

```
run:
Sum from 1 to 10 is 55
Sum from 12 to 22 is 187
Sum from 32 to 40 is 324
BUILD SUCCESSFUL (total time: 0 seconds)
```

Method assists to avoid repeating codes

```
public class KHOJ {

    public static void sumFromTo(int start , int end) {
     int sum = 0;
     for (int i = start ; i <= end ; i++){
         sum += i;
     }//End for
     System.out.println("Sum from " + start + " to " + end + " is " + sum);
    } // End sumFromTo method

    public static void main(String[] args) {
     sumFromTo(1  , 10);
     sumFromTo(12 , 22);
     sumFromTo(32 , 44);
    } // End Main Method

} // End Class
```

```
run:
Sum from 1 to 10 is 55
Sum from 12 to 22 is 187
Sum from 32 to 40 is 324
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exercise: sum loops method – return no value

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

    sumVoid(1   , 10);
    sumVoid(12  , 22);
    sumVoid(32  , 44);

}
```

```
public static void sumVoid(int start,int end){
  int sum = 0;
  for (int i = start; i <= end; i++) {
        sum += i;
  }
 System.out.println("Sum from "+start+" to "+end+"is "+ sum);
}
```

Exercise: sum loops method – return int value

```
public static void main(String[] args){
  System.out.println("Sum from 1 to 10 is  "
+ sumInt(1, 10));
  System.out.println("Sum from 12 to 22 is "
+ sumInt(12, 22));
  System.out.println("Sum from 32 to 44 is "
+ sumInt(32, 44));
 }
```

```
public static int sumInt(int start,int end){

    int sum = 0;
    for (int i = start; i <= end; i++) {
       sum += i;
    }
    return sum;
}
```

Passing Parameters

```
public class KHOJ {

    public static void nPrintln(String message, int n){
     for(int i=0 ; i<n ; i++)
          System.out.println(message);
    } //End nPrintln Method

    public static void main(String[] args) {
     nPrintln("Ahmad Khoj" , 5);

     nPrintln("COCS 202" , 15);
    } // End Main Method

} // End Class
```

```
run:
Ahmad Khoj
Ahmad Khoj
Ahmad Khoj
Ahmad Khoj
Ahmad Khoj
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
COCS 202
```

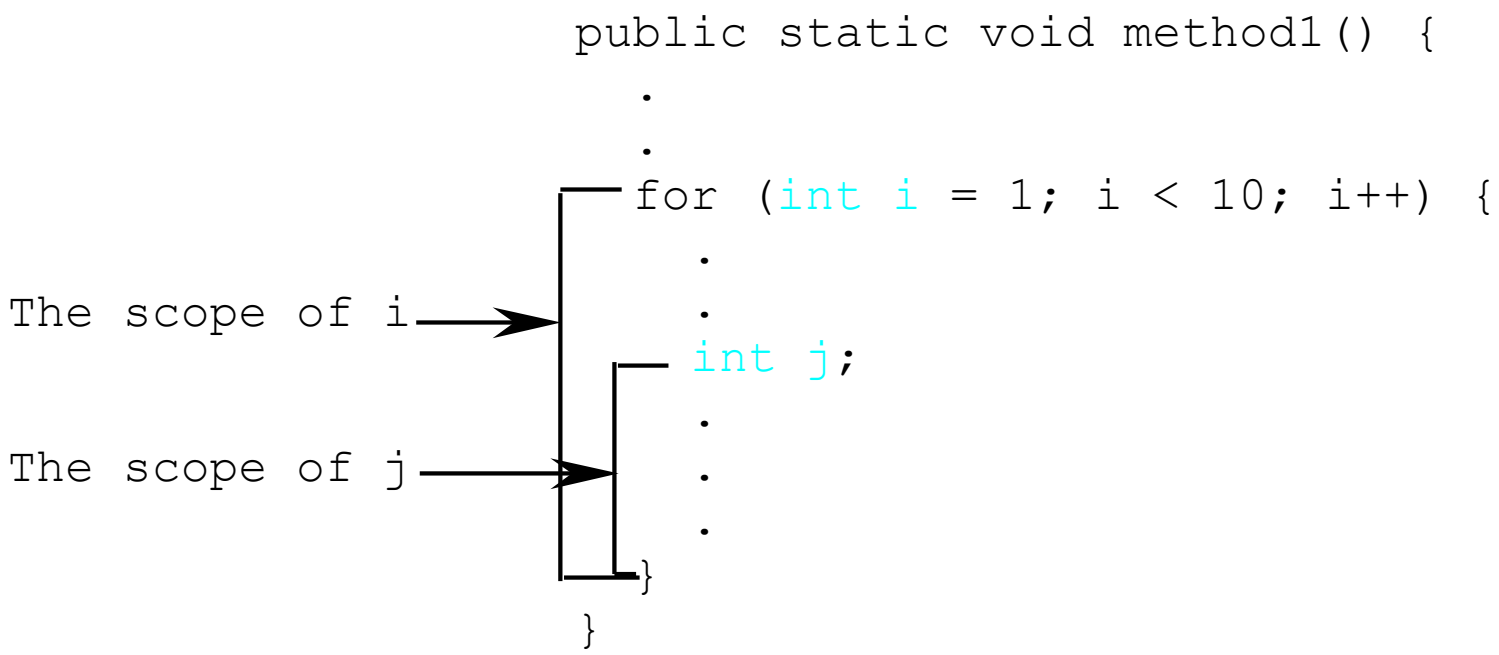
Scope of Local Variables

- A local variable: a variable defined inside a method.
- Scope: the part of the program where the variable can be referenced.

- The scope of a local variable starts from its declaration and continues to the end of the block that contains the variable.
- A local variable must be declared before it can be used.

Scope of Local Variables, cont.

- A variable declared in the initial action part of a `for` loop header has its scope in the entire loop.
- But a variable declared inside a `for` loop body has its scope limited in the loop body from its declaration and to the end of the block that contains the variable.



Scope of Local Variables, cont.

- It is fine to declare i in two non-nesting blocks

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

     int x = 0;
     int y = 0;

     for(int i=1 ; i<=10 ; i++){
         x+=i;
     }

     for(int i=1 ; i<=10 ; i++){
         y+=i;
     }

        System.out.println("x= " + x);
        System.out.println("y= " + y);
    } // End Main Method
} // End Class
```

- It is wrong to declare i in two nesting blocks

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

     int sum = 0;

     for(int i=1 ; i<=10 ; i++){
     for(int i=1 ; i<=10 ; i++)
         sum+=i;
     } //end for #1

        System.out.println("sum= " + sum);
    } // End Main Method
} // End Class
```

The `Math` Class

- Class constants:
  - `PI`

```
Math.PI; //returns 3.141592653589793
```

- Class methods:
  - Trigonometric Methods
  - Exponent Methods
  - Rounding Methods
  - min, max, abs, and random Methods

Trigonometric Methods

- sin(double a)
- cos(double a)
- tan(double a)

```
Examples:

Math.sin(0) //returns 0.0
Math.sin(Math.PI/ 6) //returns 0.5
Math.sin(Math.PI/ 2) //returns 1.0

Math.cos(0) //returns 1.0
Math.cos(Math.PI/6) //returns 0.866
Math.cos(Math.PI / 2) //returns 0
```

Exponent Methods

- `pow(double a, double b)`

  Returns `a` raised to the power of `b`.

```
Math.pow(2, 3);   //returns 8.0
```

- `sqrt(double a)`

  Returns the square root of `a`.

```
Math.sqrt(4);     //returns 2.0
```

Rounding Methods

- `double ceil(double x)`

  x rounded up to its nearest integer. This integer is returned as a double value.

- `double floor(double x)`

  x is rounded down to its nearest integer. This integer is returned as a double value.

- `int round(float x)`

  Return (int)Math.floor(x+0.5).

Rounding Methods Examples

```
Math.ceil(2.1) //returns 3.0
Math.ceil(2.0) //returns 2.0

Math.floor(2.1) //returns 2.0
Math.floor(2.0) //returns 2.0

Math.round(2.5) //returns 3.0
Math.round(2.4) //returns 2.0
```

random Method

Generates a random double value greater than or equal to 0.0 and less than 1.0.

Examples:

```
Math.random();
```
→ Returns a random double between 0.0 and 1.0

```
(int)(Math.random() * 10);
```
→ Returns a random integer between 0 and 9.

The Math Class

```
//*** Class constants ***
  double a = Math.PI; //3.141592653589793
//*** Class methods ***
//Trigonometric Methods
  double b = Math.sin(0); //0.0
  double c = Math.cos(Math.PI/6); //0.8660254037844387
  double d = Math.tan(Math.PI/2); //1.6331239353195370E16
//Exponent Methods
  double e = Math.pow(2, 3); //8.0
  double f = Math.sqrt(4); //2.0
//Rounding Methods
  double g = Math.ceil(7.1); //8.0
  double h = Math.floor(7.9); //7.0
  double i = Math.round(7.5); //8.0
  double j = Math.round(7.4); //7.0
//min, max, abs, and random Methods
  double k = Math.min(10, 5); //5.0
  double l = Math.max(10, 5); //10.0
  double m = Math.abs(-10); //10.0
  double n= Math.random(); //from 0.0 to 1.0
     int o= (int) (Math.random() * 10); //from 1 to 10;

     System.out.println("PI= "+a+"\nsin= "+b+"\ncos= "+c+"\ntan= "+
                        d+"\npow= "+e+"\nsqrt= "+f+"\nceil= "+
                        g+"\nfloor= "+h+"\nround= "+i+"\nround= "+
                        j+"\nmin= "+k+"\nmax= "+l+"\nabs= "+
                        m+"\nrandom= "+n+"\nrandom= "+o);
```

```
run:
PI= 3.141592653589793
sin= 0.0
cos= 0.8660254037844387
tan= 1.6331239353195370E16
pow= 8.0
sqrt= 2.0
ceil= 8.0
floor= 7.0
round= 8.0
round= 7.0
min= 5.0
max= 10.0
abs= 10.0
random= 0.2270823610314524
random= 8
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Create the following methods to calculate area of (Triangle and Rectangle):**

  1. **public static void triangleArea (double base , double height)**
  2. **public static double recArea(double length , double width)**

- **What you should have inside each method:**

  1. **triangleArea method:** calculate triangle area using this formula:

     triangle = 0.5 * base * height

  2. **recArea method:** calculate rectangle area using this formula:

     rectangle = length * width

- **Main**
  - **Make decision by using switch to choose which area will be calculated**
  - **When user chooses one, prompt him to enter base and height for triangle**
  - **When user chooses two, prompt him to enter length and width for rectangle**
  - **When user chooses other numbers, display this message "Wrong Choice"**

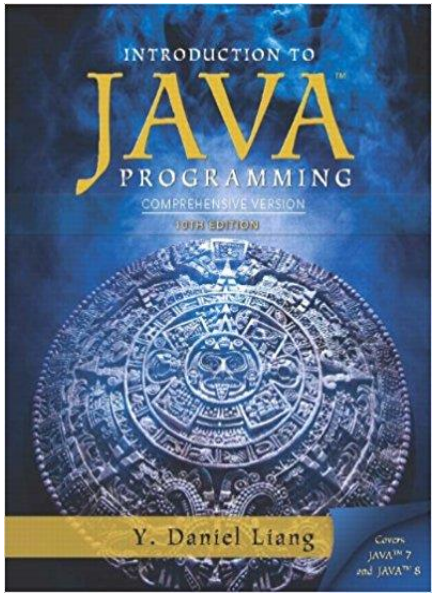
```java
import java.util.Scanner;
public class KHOJ {
    public static void triangleArea(double base, double height){
        double triangle = 0.5 * base * height;
        System.out.println("The area of triangle is " + triangle);
    }//end triangleArea

    public static double recArea(double length, double width){
        double rectangle = length * width;
        return rectangle;
    }//end recArea

    public static void main(String[] args) {
        Scanner input = new Scanner( source:System.in);
        System.out.print( s:"Enter 1 to calculate area of Triangle or 2 to calcuate area of rectangle: ");
        int ch = input.nextInt();

        switch(ch){
            case 1:
                System.out.print( s:"Enter base: " );
                double base = input.nextDouble();
                System.out.print( s:"Enter height: " );
                double height = input.nextDouble();
                triangleArea(base, height);
                break;
            case 2:
                System.out.print( s:"Enter length: " );
                double length = input.nextDouble();
                System.out.print( s:"Enter width: " );
                double width = input.nextDouble();
                System.out.println("The area of rectangle is " + recArea(length, width) );
                break;
            default:
                System.out.println( x:"Wrong choice");
        }//end switch
    }//end main
}//end class
```

Introduction to Java Programming
(10th Edition)



By
Y. Daniel Liang