

Return statement

The `return` statement is used in a function body and tells Java to exit the function. Depending on whether the function returns a value or not, the syntax of the `return` statement will be one of the following:

```
return <value>; //in a function that does return a value
return; //in a function that does not return a value
```

Consider the following function. It returns the first order from a given client found in the collection `orders`:

```
Order getOrderFrom( Person client ) { //the function return type is Order
    if( orders == null )
        return null; //if the collection orders does not exist, return null
    for( Order o : orders ) { //for each order in the collection
        if( o.client == client ) //if the field client of the order matches the given client
            return o; //then return that order and exit the function
    }
    return null; //we are here if the order was not found - return null
}
```

If the `return` statement is located inside one or several nested loops or if statements, it immediately terminates execution of all of them and exits the function. If a function does not return a value, you can skip the `return` at the very end of its body: the function will be exited in its natural way:

```
void addFriend( Person p ) { //void means no value is returned
    if( friends.contains( p ) )
        return; //explicit return from the middle of the function
    friends.add( p );
    //otherwise the function is exited after the last statement - no return is needed
}
```