Methods

IST 256

Application Programming for Information Systems

Concept of Procedures and Functions

- All modern programming languages have a way to break the program into smaller pieces by placing a piece of code in a procedure or function
  - This type of modularization is good programming design

  - Often entire sections of code need to be repeated in different parts of the program, for example sorting a list of data, and making it into a procedure streamlines the process
    - Write once, use many times
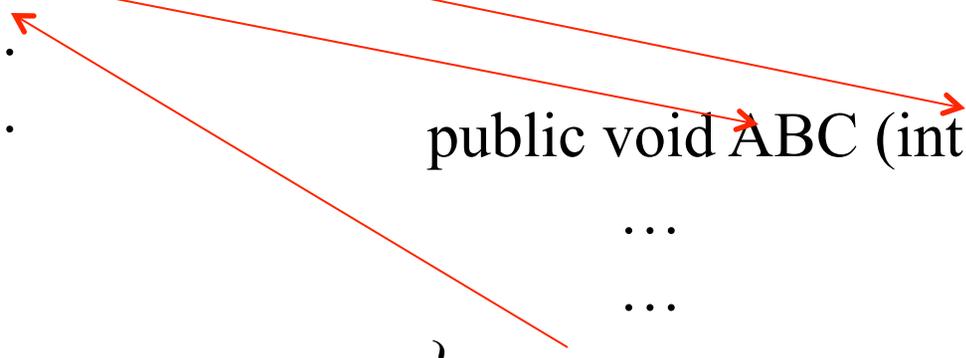
- Java calls these **methods**

Use methods to organize tasks

- Suppose you have a task called ABC; use a method to contain the task code:

```
public static void main(String[] args) {
        int number;
    …
    …
        ABC (number)
    …
    …
}
                                public void ABC (int targetnum) {
                                        …
                                        …
                                }
```

When the main method wants to compute this task, it "calls" ABC, which causes execution to jump to the code of the task

Organize code into methods

```
public static void main(String[] args) {
        String query = "???";
        String docs, results;

        ……
        docs = search (query);
        results = rank (docs);
        show_results(results);

        …

        …
}
```

```
private String search (String query) {
        String docs;

        …
        return docs;

}
```

4

Method definition

scope keywords   return type*   method name   formal parameter list

method
header

public static int squareR (int y) {

method
body

// variable to hold the square of y
int ysquared;

ysquared = y * y;
return ysquared;
}
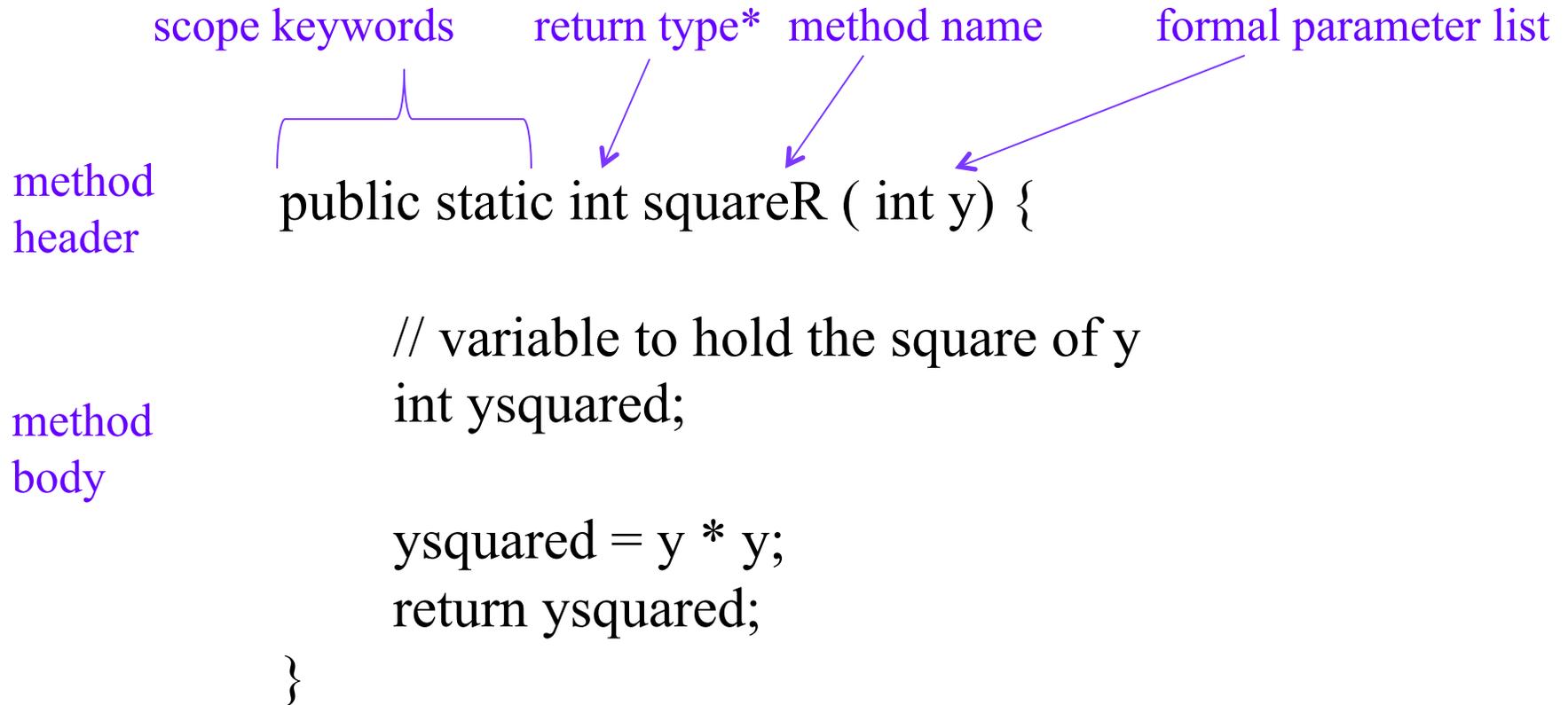
* The return type may be the keyword *void*, if there is no value to return.

5

Method call

- The method call gives an actual parameter value, which is copied to be the value of the formal parameter

int result;

// save the result of the method

result = squareR (33);

6

Flow of control for method calls

int result;

// save the result
result = squareR (33);

public static int squareR (int y) {

// variable to hold the square of y
int ysquared;

ysquared = y * y;
return ysquared;

}