Chapter 5

Calling Methods

Method call

```
f();
```

`main` calls `f`

```
 1 class Call1
 2 {
 3    public static void main(String[] args)
 4    {
 5       System.out.println("before call");
 6       f();
 7       System.out.println("after call");
 8    }
 9    //---------------------------------
10    public static void f()
11    {
12       System.out.println("in f");
13    }
14 }
```

Parameters

```
 1 class Call2
 2 {
 3    public static void main(String[] args)
 4    {
 5       int x = 1;
 6       displaySum(x, x + 5, 20);
 7       System.out.println("All done");
 8    }
 9    //-----------------------------------
10    public static void displaySum(int a, int b, int
c)
11    {
12       int sum;
13       sum = a + b + c;
14       System.out.println("sum = " + sum);
15    }
16 }
```

What happens during a method call

1) The parameters `a`, `b`, and `c` are created

2) The values of the arguments—1, 6, and 20—are **passed** to their corresponding parameters. That is, 1, 6, and 20 are automatically assigned to `a`, `b`, and `c`, respectively.

3) The local variable `sum` is created (line 12). It has no initial value.

4) The body of the `displaySum` method is executed. Line 13 computes the sum of `a`, `b`, and `c`, and assigns the result to `sum`. Line 14 displays the value in `sum`.

5) The local variable `sum` and the parameters `a`, `b`, `c` are destroyed.

6) Control returns to the statement that follows the call of the `displaySum` method (line 7).

Local variables and parameters have local scope

```
 1 class Call3
 2 {
 3    public static void main(String[] args)
 4    {
 5       int x = 1;
 6       displaySum(x, x + 5, 20);
 7       System.out.println("All done");
 8    }
 9    //---------------------------------
10    public static void displaySum(int x, int y, int z)
11    {
12       int sum;
13       sum = x + y + z;
14       System.out.println("sum = " + sum);
15    }
16 }
```

Argument-parameter type mismatch

```
public static void g(int i) is called with

g(20); // okay

g(d);  // illegal if d is double

g((int)d);  // okay
```

Returning a value

```
 1 class ReturnValue1
 2 {
 3    public static void main(String[] args)
 4    {
 5       int x = 1, y;
 6       y = getSum(x, x + 5, 20);
 7       System.out.println("sum = " + y);
 8       System.out.println("All done");
 9    }
10    //---------------------------------
11    public static int getSum(int a, int b, int c)
12    {
13       int sum;
14       sum = a + b + c;
15       return sum;        // returns value in sum
16    }
17 }
```

```
 1 class ReturnValue2
 2 {
 3    public static void main(String[] args)
 4    {
 5       int x = 1
 6       System.out.println("sum = " + getSum(x, x + 5, 20));
 7       System.out.println("All done");
 8    }
 9    //---------------------------------
10    public static int getSum(int a, int b, int c)
11    {
12       return a + b + c;   // both computes and returns sum
13    }
14 }
```

Overloading method names

```
 1 class Overloading
 2 {
 3    public static void main(String[] args)
 4    {
 5       o();         // calls o on line 12
 6       o(2);        // calls o on line 17
 7       o(2.0);      // calls o on line 22
 8       o(2, 2.0);   // calls o on line 27
 9       o(2.0, 2);   // calls o on line 32
10    }
```

```
11    //-----------------------------------
12    public static void o()
13    {
14       System.out.println("No args");
15    }
16    //-----------------------------------
17    public static void o(int i)
18    {
19       System.out.println("i = " + i);
20    }
21    //-----------------------------------
```

```
22    public static void o(double d)
23    {
24      System.out.println("d = " + d);;
25    }
26    //-----------------------------------
27    public static void o(int i, double d)
28    {
29       System.out.println("i = " + i + "   d = " + d);
30    }
31    //-----------------------------------
32    public static void o(double d, int i)
33    {
34       System.out.println("d = " + d + " i = " + i);
35    }
36 }
```

External and internal calls and accesses

```
 1 class One
 2 {
 3    public static void main(String[] args)
 4    {
 5       Two.add();                              // external call
 6       System.out.println("x = " + Two.x);  // external access
 7    }
 8
 9 }
10 //=============================================
11 class Two
12 {
13    public static int x = 1;     // x is a static variable
14    //---------------------------------
15    public static void add()
16    {
17       int y = 2;                // y is a local variable
18       x = x + y;                // internal access of x
19       display();                // internal call
20    }
21    //---------------------------------
22    private static void display()
23    {
24       System.out.println("x = " + x); // internal access of x
25    }
26 }
```