Pass (Call) by Value and by Reference

```
class Account {

    double balance = 1000;
}

public class Test {

    static void test(double v, Account a) {
        v = v + 100.0;
        a.balance = a.balance + 100.0;
    }

    public static void main(String[] args) {

        Account a = new Account();
        double value = 1000.0;

        System.out.println("Before: Value=" + value + " Balance=" + a.balance);
        Test.test(value, a);
        System.out.println("After: Value=" + value + " Balance=" + a.balance);
    }
}
```

Passing References

```
public class Test {

    public static void method(Test a) {

        System.out.println("a inside method(): " + a);
    }

    public static void main(String[] args) {

        Test t = new Test();

        System.out.println("t inside main(): " + t);

        method(t);
    }
}
```

Aliasing

```
public class Test {

    private int i;

    public Test(int ii) {

        i = ii;
    }

    public static void main(String[] args) {

        Test x = new Test(7);
        // Assign the reference to
        // another local variable (alias).
        Test y = x;

        System.out.println("x: " + x.i);
        System.out.println("y: " + y.i);
        System.out.println("Incrementing x");

        x.i++;

        System.out.println("x: " + x.i);
        System.out.println("y: " + y.i);
    }
}
```

Method Overloading [Überladen]

```
class Parent {

    public void method() {
        System.out.println("Test Parent");
    }
}

public class Child extends Parent {

    // The same method name, but different parameter types.
    public void method(int i) {
        System.out.println("Test Child 1");
    }

    // Yet another method name in the same class, with different number of parameters.
    public void method(int i, int j) {
        System.out.println("Test Child 2");
    }

    public static void main(String[] args) {
        Child c = new Child();
        c.method();
        c.method(5);
        c.method(5, 6);
    }
}
```

Method Overriding [Überschreiben]

```
class Parent {

    public void method() {
        System.out.println("Test Parent");
    }
}

public class Child extends Parent {

    public void method() {
        // If functionality of super class's „method” is needed,
        // it has to be called explicitly with:
        // super.method();
        System.out.println("Test Child");
    }

    public static void main(String[] args) {
        Child c = new Child();
        c.method();
    }
}
```