Accessing Parent Members

- if a method overrides one of its parent class's methods, one can invoke the overridden method through the use of **super** (so we don't loose access to it)
- one can also use **super** to refer to a shadowed field present in the parent class
- when used as a constructor name during instantiation, **super** refers to a particular constructor of the parent
- in Java, invocation of a super class constructor has to be the first line in the sub-class constructor

```java
class Parent {
  String name;
  Parent() {}

  Parent(String name) {
    this.name = "Parent Attribute " + name;
  }

  void saySomething() {
    System.out.println("From ParentMethod.");
} }
```

```java
class Child extends Parent {
  String name;

  Child(String name) {
    super(name);
    this.name = "Child Attribute " + name;
  }

  String getParentName() {
    return super.name; }

  void saySomething() {
    System.out.println("From ChildMethod.");
  }

  void delegate() {
    super.saySomething();
} }
```

call constructor of parent

access shadowed attribute of parent

overriding

access parent method

```java
class SuperWorld {
  public static void main (String[] args) {
    Child  a = new Child("NameA");
    Parent b = new Parent("NameB");
    a.saySomething();
    b.saySomething();
    a.delegate();
    System.out.println(a.name);
    System.out.println(a.getParentName());
} }
```