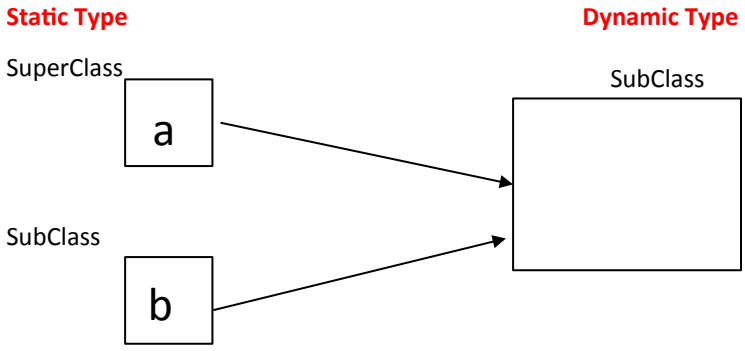
Inheritance

SuperClass a = new SubClass();
SubClass b = a;



Inheritance

Constructors:

There is **always** an implied call to the superclass constructor on the **FIRST LINE.**

```
class Child extends Parent {
    public Child() {
        System.out.println("hi!");
    }
}
```

```
class Child extends Parent
{
    public Child() {
        super();

System.out.println("hi!");
    }
}
```

these are the same!

(side note: the explicit call to super() can only be in the

Inheritance

Dynamic Method Lookup:

Suppose we have the code cat.eat(). How do we determine what this does?

1. Does the **static type** of cat have the method eat()? If no, compiler error.
2. If yes, check the **dynamic type** of cat to see if eat() is **overridden**.
   If overridden, run the **dynamic class's method**. If not overridden, run the **static class's method**.

overridden methods must have the same **signature** (method name, argument types)

Field Shadowing:

We always consider the **static type** for looking up attributes
(for example, cat.name will look at the name in cat's static type)

Inheritance

Polymorphism:

1. If the static type is a subclass of the dynamic type, COMPILE-TIME ERROR
   ex: Cat c = new Animal();
2. We can **cast** "up" to any superclass without any problems
   ex: Cat c = new Cat();
   ((Animal) c) is a valid cast.
3. We can only cast "down" so far as the object's dynamic type
   --> if we cast "down" to a subclass below the dynamic type, RUN-TIME ERROR
   ex: Animal a = new Cat();
   ((Cat) a) is a **valid** cast.
   ex: Animal a = new Animal();
   ((Cat) a) will give us a **run-time error**!!

(inspiration from Summer '12, MT1)

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Superhero superhero = new
Superhero();
superhero.punch(superhero);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Batman batman = new Batman("I'M
BATMAN!");
batman.punch(batman);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Batman batman = new
Superhero();
batman.punch(batman);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Superhero superhero = new
Batman();
superhero.punch( (Batman)
superhero);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Batman batman = new Batman();
((Superhero)
batman).punch(batman);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Superhero superhero = new
Superhero();
superhero.punch( (Batman)
superhero);
```

Inheritance

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }

        public void punch(Superhero a) {
            System.out.println("BOOM " + s);
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Superhero superhero = new
Batman();
superhero.punch( (Batman)
superhero);
```

Inheritance

NOTE the changed source code!

```
public class Superhero {
        String s;
        public Superhero() {
            s = "I'M A SUPERHERO";
            System.out.println(s);
        }

        public void punch() {
            System.out.println("Punch! Punch!");
        }
    }
```

```
public class Batman extends Superhero {
    String s;
    public Batman() {
        s = "NANANANANANANA";
    }

    public Batman(String s) {
        this.s = s;
        System.out.println(this.s);
    }

    public void punch(Superhero v) {
        s = "BATMAN!";
        super.punch(v);
        System.out.println("BOOM " + s);
    }

    public void punch(Batman b) {
        System.out.println("Wat.");
    }
}
```

```
Batman batman = new Batman();
((Superhero)
batman).punch(batman);
```