Page 6: complex numbers

A complex number is an expression of form

\[ a + bi \]

where \( a, b \in \mathbb{R} \) and \( i^2 = -1 \)

Complex arithmetic:

add, sub, ...
Multiply:

\[(a+bi)(c+di) = ac + bci + adi + bd i^2\]

\[= (ac - bd) + (bc + ad) i\]

Conjugate:

\[\overline{a+bi} = a - bi\]
Note: if \( z \in \mathbb{C} \)

say \( z = a + bi \), then

\[
\overline{z^2} = (a + bi)(a - bi) = a^2 + b^2
\]

**modulus:**

\[
|z|^2 = (\overline{z}z)^{1/2} = \sqrt{a^2 + b^2}
\]

\( \text{Re}(z) = \text{Re}(a + bi) = a \)

\( \text{Im}(z) = \text{Im}(a + bi) = b \)

**argument**

\[
\text{arg}(z) = 0 \\
\text{arg}(\overline{z}) = -\theta
\]
Reciprocal: \( \overline{z} = a + bi \)

\[
\frac{1}{z} = \frac{\overline{z}}{z \overline{z}}
\]

\[
= \frac{a - bi}{a^2 + b^2}
\]

\[
= \left( \frac{a}{a^2 + b^2} \right) + \left( \frac{-b}{a^2 + b^2} \right)i
\]

Divide

\[
\frac{w}{z} = \frac{w \overline{z}}{z \overline{z}} = \frac{(c + di)(a - bi)}{a^2 + b^2}
\]

\[
= \left( \frac{ac + bd}{a^2 + b^2} \right) + \left( \frac{ad - bc}{a^2 + b^2} \right)i
\]
Chapter 7: Inheritance

Recall all classes are sub-classes of Object

```
Object
  ↓
Person  Rational  Complex
  ↓
String  Scanner
```

We can create subclasses of existing classes by use of extends.
Ex.

```java
// Foo.java
class Foo {
}

// Bar.java
class Bar extends Foo {
}
```

creates:

```
Object
  |
  Foo
  |
  Bar
```

We can create variables of type `Foo` or `Bar`, but we can also create `Objects` of these types.
Ex - In some method in some class:

```
Foo f = new Foo();
Bar b = new Bar();
```

Any member variable or method belonging to `f` also belongs to `b`, but not conversely.
In other words:

"a Bar is a Foo"

A Foo is not a Bar however.

Subtype Principle:

A subclass object can always be used where an object of its superclass is expected. (reverse is false)

In particular:

an object of the subtype can be referred to by a variable of the supertype. (reverse is false)
So:
reference variables can point down lines of ancestry, not up.

Ex:

```java
Foo f = new Foo();
Bar b = new Bar();
```

![Diagram showing the relationship between Foo and Bar objects.]

```
f = b; // ok, subtype principle.
```

However:

```
b = f; // won't compile
```
Ex. Foo subclasses Bar.

Ex. Student