Class Fields

Recall:

- An instance field is a component of an object.
  - Since each instance of an object will possess these fields.

- In the class:

class fraction{

int num;

int den;

}

, the instance fields are num and den.

Class fields

- Class fields are used for quantities that can be shared by the members of the class because they do not change.

- Consider a class iPhone. We have fields to keep track of serial number, colour, and price.

- Each instance of iPhone will have a different serial number and colour.
- Although the price of an iPhone does change (especially when they become outdated), the price is almost usually the same for all units regardless of where you make your purchase (provided they have the same specs).
  - Therefore, we do not need to keep a copy of price for every instance of iPhone.
  - We can keep price as a class field so there is only one copy for all instances of iPhone.

For example:

```
class iPhone
{
    private int serial;
    private String colour;
    private static double price;
}

// again, since these fields are private, we cannot
   refer to them outside of the class (e.g. i.price)
```

- That is, 1149.00 will be the value for *price* for all objects of type iPhone

```
class iPhone
{
    private int serial;
    private String colour;
    private static double price = 1149.00;
}
```

What's different here?

```
class iPhone
{
    private int serial;
    private String colour;
    static double price = 1149.00;
}
```

- **If the instance field *price* is later changed for one object of type iPod, it is automatically changed for all other objects of type iPod**

  iPhone unit = new iPhone();

  unit.price = 999.00;