Class Methods

Recall:

- An instance method can only be called from an instance of a class.

- Class methods can be called without creating an instance of that class.
  - Class methods unaware of instance members
  - Declared as static.

Some sample headers:

- Instance Method

```
public int larger(){….
}
```

- Class Methods

```
public static int larger (int x, int y){ ….
}
public static void larger (int x, int y){ ….
}
```

- With objects, it is possible to use class methods (i.e. classes with static in them)

- We could define the functionality of instance and class methods by the following……..

Instance Methods

- controls the functionality of the object.
  - Instance methods does something to the objects.
- Instance methods will need to be called using an (implicit) object that has already been created
- Instance methods will have implicit fields and refer to an implicit object.

Class methods

- Does something with objects
- There are no implicit objects or fields
  - All objects being dealt with are explicit and are treated as ordinary pararmeters
  - Are invoked using the name of the class instead of the object.
- Class methods are ordinary methods that are associated with the class definition of an object

- In your object fraction, add the class method:

```
public static fraction largerfraction (fraction f1, fraction f2){
// This method will receive two parameters (f1, f2) of type
   fraction and return the larger of the two fractions as a
   type fraction.
}
```

- In the main method the call would be:

```
fraction fresult = new fraction();
fraction ffirst = new fraction(1,2);
fraction fsecond = new fraction(2,3);
fresult = fraction.largerfraction (ffirst, fsecond);
```

NOTICE:

- notice there are NO implicit objects.
- When calling a class method we precede the method name with the class identifier (i.e. fraction) instead of an object.

- The call will follow the following template:

<class identifier>.<method identifier>(<parameter list>);

For example

fresult = fraction. largerfraction (ffirst, fsecond);

Remember:

- We are simply calling a method that happens to be in a class definition.
- We do not use an implicit object to invoke class methods therefore:
  - DO NOT ACCESS THE CLASS FIELDS OF YOUR OBJECT DIRECTLY!!!
  - USE YOUR ACCESSOR/MUTATOR METHODS!