Class Fields and Methods

1. What is the difference between the declaration of a class method and that of an instance method?

An instance method is without the “static” keyword, and to access the instance method you would require a reference object. Instance methods can change instance fields and class variables. An instance field differs from a class method, is created with the “static” keyword to allow it to run without an object to reference to, although it may require a the class name to run. It can change the

1. Assume that methods in a class have been declared with the public modifier. From outside that class, how would we refer to
   1. a class method?

<Class>.<method>(<paremeters>);

* 1. an instance method?

<Class> <objectname> = new <Class>(<paremeters>);

<objectname>.<method>(<paremeters>);

1. Write definitions for the following class methods that could be used in the Fraction class.
   1. The method product should have two Fraction parameters. It should return a value of type Fraction, the product of the parameters passed to it.

| public static Fraction product(Fraction f1, Fraction f2)  {  Fraction returnValue = new Fraction();  returnValue.numerator = f1.numerator\*f2.numerator;  returnValue.denominator = f1.denominator\*f2.denominator;  return returnValue;    } |
| --- |

* 1. The method abs should have a single Fraction parameter. It should return a value of type Fraction in which any negative fields in the parameter have been replaced by their absolute values.

| public static Fraction abs(Fraction f1)  {  Fraction returnValue = new Fraction();  returnValue.numerator = Math.abs(f1.numerator);  returnValue.denominator = Math.abs(f1.denominator;  return returnValue;    } |
| --- |

* 1. The method isPositive should have one Fraction parameter. It should return a boolean value: true if its parameter represents a positive fraction and false otherwise.

| public static boolean abs(Fraction f1)  {  if (f1.numerator \* f1.denominator > 0)  {  return true;  }  else  {  return false;  }    } |
| --- |

1. What is the difference between the declaration of a class field and that of an instance field?

Instance field data values can only be changed by instance methods, which require an object (non-static). However, class fields can be declared, set, and retrieved with instance and class methods, as they are non-static.

1. Assume the fields in a class have been declared with the public modifier. From outside that class, how would we refer to
   1. a class field?

<ClassName>.<variableName>

* 1. an instance field?

//It is required to create an object for non-static methods:

<Class> objEx = new <Class>(<Parameters>);

objEx.<variableName>