

Program 2: Specifications

public class **Fraction**

The Fraction class wraps the values of two ints into an object representing a fraction. An object of type Fraction contains two fields whose type is int -- n and d.

In addition, this class provides several methods for implementing operations between two Fractions, as well as other methods useful when dealing with a Fraction.

Field Summary	
Type	Field
private int	d An int representing the numerator of the fraction, holding the maximum value an int can have, $2^{31}-1$.
private int	n An int representing the denominator of the fraction, holding the maximum value an int can have, $2^{31}-1$.

Constructor Summary	
Fraction() Constructs a newly allocated Fraction object, with a value of 0 for the numerator and a value of 1 for the denominator.	
Fraction(int num, int den) Constructs a newly allocated Fraction object, with the value of num as the numerator and the value of den as the denominator. It returns a simplified fraction.	

Method Summary	
Return Type	Method
Fraction	add(Fraction frac) Adds the Fraction <i>this</i> to the Fraction frac and returns the resulting simplified Fraction.
int	compareTo(Fraction frac) Compares the value of <i>this</i> Fraction to the value of frac. If <i>this</i> is bigger, it returns an int value of 1; if <i>this</i> is smaller, it returns an int value of -1; if the two Fractions are of equal value, it will return an int value of 0.
Fraction	divide(Fraction frac) Divides the Fraction frac into <i>this</i> and returns the resulting simplified Fraction.
int	getDen() Returns the int value of the denominator in the Fraction <i>this</i> .
int	getNum() Returns the value of the numerator in the Fraction <i>this</i> .
Fraction	multiply(Fraction frac) Multiplies the Fraction <i>this</i> with the Fraction frac and returns the resulting simplified Fraction.
Fraction	reciprocal() Calculates and returns the reciprocal of the Fraction <i>this</i> .

void	setDen(int den) Assigns the value of den to the denominator of the Fraction <i>this</i> . If the value of the den is 0, it still assigns it. However, it will print a message to the screen saying the fraction is invalid.
void	setNum(int num) Assigns the value of num to the numerator of the Fraction <i>this</i> .
Fraction	simplify() Simplifies the Fraction <i>this</i> . If the denominator is 0, it prints an error message to the screen
Fraction	subtract(Fraction frac) Subtracts the Fraction frac from the Fraction <i>this</i> and returns the resulting simplified Fraction.
String	toString() Returns a String object representing the Fraction <i>this</i> . If the numerator is 0, then it just returns "0", not "0/0".

public class **mixedNumber**

The mixedNumber class wraps an int and a Fraction object in an object. An object of type mixedNumber contains two fields whose types are int and Fraction.

In addition, this class provides several methods for implementing operations between two mixedNumbers, as well as other methods useful when dealing with a mixedNumber.

Field Summary	
Type	Field
int	whole An int representing the whole number part of the mixed number, holding the maximum value an int can have, $2^{31}-1$.
Fraction	part A Fraction representing the fraction part of the mixed number.

Constructor Summary	
mixedNumber()	Constructs a newly allocated mixedNumber object with a whole number value of 0 and a fraction of value 0/1.
mixedNumber(int int1, Fraction frac)	Constructs a newly allocated mixedNumber object, with the value of int1 as the whole number and the object frac as the fraction. It returns a simplified mixedNumber. If int1 is positive and frac is negative, frac will be subtracted from int1 and the resulting simplified mixedNumber will be returned; the opposite is also true. If both are negative or both are positive, then they are stored as is.
mixedNumber(int int1, int num, int den)	Constructs a newly allocated mixedNumber object, with the value of int1 as the whole number and a new Fraction is created using num and den as the numerator and denominator, respectively. This Fraction is stored as the fraction part of the mixed number. It returns a simplified mixedNumber.

Method Summary	
Return Type	Method
mixedNumber	add(mixedNumber numb) Adds the mixedNumbers this and numb and returns the resulting simplified mixedNumber.
int	compareTo(mixedNumber numb) Compares the value of the two mixedNumbers this and numb. Returns an int of value -1 if numb is larger, an int of value 1 if numb is smaller, and an int of value 0 if they are equal.
mixedNumber	divide(mixedNumber numb) Divides the mixedNumber this with the mixedNumber numb and returns the resulting simplified mixedNumber.
int	getDen() Returns the int value of the denominator in the Fraction part of the mixedNumber <i>this</i> .
int	getNum() Returns the value of the numerator in the Fraction part of the mixedNumber <i>this</i> .
int	getWhole() Returns the int value of the whole number part of the mixedNumber <i>this</i> .
mixedNumber	multiply(mixedNumber numb) Multiplies the mixedNumbers this and numb and returns the resulting simplified mixedNumber.
mixedNumber	reciprocal() Calculates and returns the mixed Number that is the reciprocal of the mixedNumber.
void	setDen(int den) Assigns the value of den to the denominator of the Fraction part of the mixedNumber <i>this</i> .
void	setNum(int num) Assigns the value of num to the numerator of the Fraction part of the mixedNumber <i>this</i> .
void	setWhole(int int1) Assigns the value of int1 to the whole number part of the Fraction <i>this</i> .
mixedNumber	simplify() Simplifies the mixedNumber.
mixedNumber	subtract(mixedNumber numb) Subtracts the mixedNumber numb from the mixedNumber and returns the resulting simplified mixedNumber.
String	toString() Returns a String object representing the mixedNumber. If the numerator of the fraction part of the mixedNumber <i>this</i> is zero, and for instance the whole part is 5, it will return "5", not "5 0".