(a) (2 points) Write code that declares the variable number as an `int` with the value 13.

(b) (2 points) What will be stored in the variable x:

```
double x = 3/6;
```

(c) (2 points) Explain why the following code produces a compiler error:

```
int i = 3.14;
```

(d) (2 points) Define a **character literal** and provide an example of one.

(e) (2 points) Complete the missing line of code required so that the program will display the negative of integer value entered by the user:

```
System.out.print("Enter an integer: ");
Scanner in = new Scanner(System.in);
int input = in.nextInt();
int negativeInput;
// Add your answer below this line


System.out.println("The answer is: " + negativeInput);
```

Consider the following program:

```
public static void main(String[] args) {
  String input = JOptionPane.showInputDialog(null, "Enter an integer.");
  int num = Integer.parseInt(input);
  if(num > 0) {
    JOptionPane.showMessageDialog(null, "A");
    if(num < 3) {
      JOptionPane.showMessageDialog(null, "B");
    } else {
      JOptionPane.showMessageDialog(null, "C");
    }
  } else if(num == -3) {
    JOptionPane.showMessageDialog(null, "D");
  } else {
    JOptionPane.showMessageDialog(null, "E");
  }
}
```

(a) What will be displayed if 100 is entered by the user?

(b) What will be displayed if 3 is entered by the user?

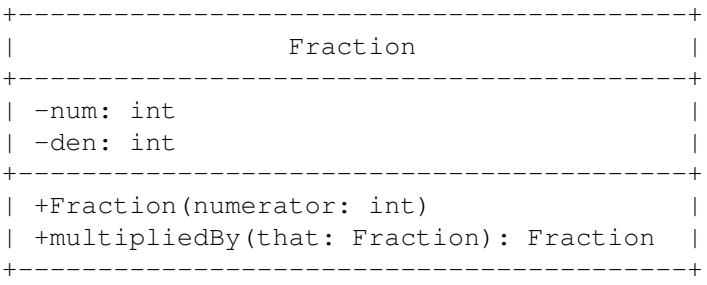
(c) What will be displayed if -3 is entered by the user?

1. (3 points) Explain why `java.lang.Double` is referred to as a wrapper class.

2. (7 points) Write a program that asks the user to enter a phrase and then displays the number of letters (upper- or lower-case) entered.

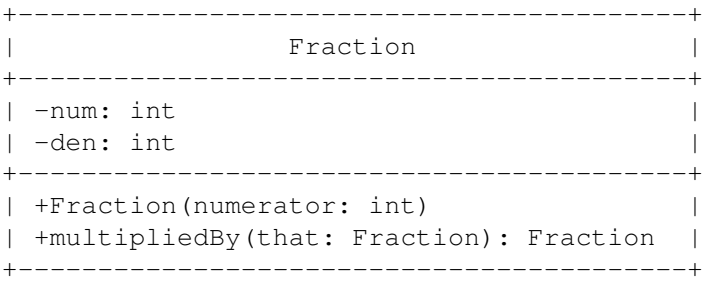
```
public class Quiz4 {
  public static void main(String[] ignored) {
```

Implement the Fraction class described in the following UML diagram:



The multipliedBy method should return a new object that is the product of the calling object and the object passed to the method. The values of the calling object and object passed to the method must not change.

Implement the Fraction class described in the following UML diagram:



The multipliedBy method should return a new object that is the product of the calling object and the object passed to the method. The values of the calling object and object passed to the method must not change.

Write a class method called sum3 that accepts an array of `int`s of length 3 and returns the sum of all the elements.

Write a class method called lucky13 that accepts an array of `int`s of any length and returns `true` if an only if the array does not contain any 1's or 3's.

Write a class method called separate that takes the strings in one list and separates them into two lists based on the lengths of the strings. The method should accept three ArrayLists of Strings, named: all, shorts, and longs. When the method is called, all, should contain at least one string while shorts and longs should be empty. When the method is done, shorts should contain all strings from all that are less than ten characters in length and longs should contain all strings from all that are at least ten characters in length.