Ira A. Fulton Schools of Engineering

school of computing, informatics, & decision systems engineering

CSE 110 – Assignment #6

Maximum points: 20 pts

Topics

- Object Oriented Programming & Methods (Chapter 5 & 8)
  - Tester class development (Command Menu)
  - Implementing instance methods

Your programming assignments require individual work and effort to be of any benefit. Every student must work independently on his or her assignments. This means that every student must ensure that neither a soft copy nor a hard copy of their work gets into the hands of another student. Sharing your assignments with others in any way is **NOT** permitted. Violations of the University Academic Integrity policy will not be ignored. The university academic integrity policy is found at_
http://www.asu.edu/studentlife/judicial/integrity.html

Use the following Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note:

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. **(If not, -1 Pt)**

```
//*******************************************************
// Name: your name
// Title: title of the source file
// Description: Write the description in your words.
// Time spent:  how long it took you to complete the assignment
// Date: the date you programmed
//*******************************************************
```

Part 1: There are no written exercises in this assignment

Part 2: Programming (20 points):

1) Write a class definition (not a program, there is no main method) named **Fraction** (saved in a file **Fraction.java**) that calculates the math operations as a fraction. A Fraction is an object with two integers, numerator and denominator. A new Fraction object starts as 0 (0/1). The test program asks to choose the operation (+, -, *, or /) and type two integers as one single line. Then, the Fraction shows the operation and result.

Your Fraction class must have:

- two instance variables
  – the **numerator** and **denominator** (Fraction = $\frac{numerator}{denominator}$)
- Must implement the following methods

Methods	Description of the methods
public Fraction (int numerator, int denominator)	Constructor-Sets the instance variables with the inputs **(2 Pts)**
public String toString()	It returns the fraction as a string such as "5/3" when the numerator is 5 and denominator 3. When the numerator is 0, it returns "0", and when the denominator is "1", it returns the numerator. 3/1 ==> "3" **(2 Pts)**
private int greatestCommonDivisor (int num1, int num2)	It reads two integers and returns the greatest common divisor by using the following algorithm. 1. Create a loop, and subtract the smaller integer from the larger one (if the integers are equal you may choose either one as the "larger") during each iteration. 2. Replace the larger integer with the computed difference. 3. Continue looping until one of the integers becomes zero. For instance, assume that two integers 24 and 18 are given. In the first iteration, the larger item is 24, and the difference is 24 – 18 = 6. In the next iteration, 24 is replaced with 6 so the larger item is set as 18, and the difference is 18 – 6= 12. In the third iteration, 18 is replaced with 12 so the larger item is set as 12, and the difference is 12 – 6 = 6. Now two items are both 6, so the difference is 0, then stop the loop and return 6. **(Not allowed to use Recursive approach: -3 Pts)**
private void simplify()	It simplifies the fraction. Case-1: When the denominator is negative, multiply -1 with both denominator and numerator. For example, 2/-3

	==> -2/3 Case-2: When the numerator and denominator are not coprime (relatively prime), divide both by their greatest common divisor. For example, 4/12 ==> 1/3 **(2 Pts)**
public **static** Fraction multiply (Fraction a, Fraction b)	It takes two Fractions, and returns the result of multiplication of them, a * b. **(1 Pts)**
public **static** Fraction divide (Fraction a, Fraction b)	It takes two Fractions, and returns the result of division of them, a / b. **(1 Pts)**
public **static** Fraction add (Fraction a, Fraction b)	It takes two Fractions, and returns the result of addition of them, a + b. **(2 Pts)**
public **static** Fraction subtract (Fraction a, Fraction b)	It takes two Fractions, and returns the result of subtraction a - b. **(2 Pts)**

2) Download and **complete (fill out the missing code)** the test driver called **Assignment6.java (5 Pts).** Assignment6.java contains the main method and creates a new Fraction object to test the methods in the class Fraction. Assignment6.java will ask a user to enter one of the following commands. Based on the user's choice, the program needs to perform corresponding operation. This will be done by calling (invoking) one of the methods you defined in the Fraction class. The program will terminate when the user enters 'q'.

```
Command Options
-----------------------------------
c: reset the value
+: add a fraction to the current value
-: subtract a fraction from the current value
*: multiply a fraction to the current value
/: divide the current value by a fraction
?: display the menu again
q: quit this program
```

Here is the description for each option:

command	task
c	resets the global Fraction as 0. (0/1)
+	asks to type two integers and construct a new Fraction. Calculate and display the value of global Fraction and the new Fraction. The result is set as the current Fraction so that the additional operation can be made.
-	asks to type two integers and construct a new Fraction. Calculate and display the value of global Fraction and the new Fraction. The result is set as the current Fraction so that the additional operation can be made.
*	asks to type two integers and construct a new Fraction. Calculate and display the value of global Fraction and the new Fraction. The result is set as the current Fraction so that the additional operation can be made.

/	asks to type two integers and construct a new Fraction. Calculate and display the value of global Fraction and the new Fraction. The result is set as the current Fraction so that the additional operation can be made.
?	displays the menu
q	quits the program

You must complete the code inside cases 'c', '+', '-', '*' and '/'' to read the needed input, call the appropriate method of the Fraction class, and display an appropriate output. A sample output is shown below.

Helpful hints for doing this assignment:

- work on it in steps – write one method, test it with a test driver and make sure it works before going on to the next method
- always make sure your code compiles before you add another method
- your methods should be able to be called in any order
- Operation of Factions: The result is also a Fraction, which needs two values of new numerator and denominator as shown below.

$$A * B = \frac{< A\ numerator >}{< A\ denominator >} * \frac{< B\ numerator >}{< B\ denominator >}$$

$$= \frac{< A\ numerator > * < B\ numerator >}{< A\ denominator > * < B\ denominator >} = \frac{new\ numerator}{new\ denominator}$$

$$A + B = \frac{< A\ numerator >}{< A\ denominator >} + \frac{< B\ numerator >}{< B\ denominator >}$$

$$= \frac{< A\ numerator > * < B\ denominator > + < B\ numerato > * < A\ denominator >}{< A\ denominator > * < B\ denominator >}$$

Sample Output (user input is in red, which will not seen online submission):

```
----------------
INPUT 1
----------------
c
+
3 5
+
1 2
-
2 3
*
4 7
c
```

```
+
4 2
/
12 8
?
q

----------------
YOUR OUTPUT 1
----------------
*** Start of Program ***

Command Options
-----------------------------------
c: reset the value
+: add a fraction to the current value
-: subtract a fraction from the current value
*: multiply a fraction to the current value
/: divide the current value by a fraction
?: display the menu again
q: quit this program

Value:0

[Please enter a command or type ?] c
Value:0

[Please enter a command or type ?] +
[Enter two numbers for a fraction to ADD] 3 5
0 + 3/5 = 3/5
Value:3/5

[Please enter a command or type ?] +
[Enter two numbers for a fraction to ADD] 1 2
3/5 + 1/2 = 11/10
Value:11/10

[Please enter a command or type ?] -
[Enter two numbers for a fraction to SUBTRACT] 2 3
11/10 - 2/3 = 13/30
Value:13/30

[Please enter a command or type ?] *
[Enter two numbers for a fraction to MULTIPLY] 4 7
13/30 * 4/7 = 26/105
Value:26/105

[Please enter a command or type ?] c
Value:0

[Please enter a command or type ?] +
[Enter two numbers for a fraction to ADD] 4 2
0 + 4/2 = 2
Value:2

[Please enter a command or type ?] /
[Enter two numbers for a fraction to DIVIDE]  12 8
```

```
2 / 12/8 = 4/3
Value:4/3

[Please enter a command or type ?] ?

Command Options
-----------------------------------
c: reset the value
+: add a fraction to the current value
-: subtract a fraction from the current value
*: multiply a fraction to the current value
/: divide the current value by a fraction
?: display the menu again
q: quit this program

Value:4/3

[Please enter a command or type ?] q
*** End of Program ***
```

/**

Submit your homework by following the instructions below:

**/

- Go to the course web site (my.asu.edu), and then click then click on the GradeScope on CANVAS.
- Submit the **Assignment6.java** and **Fraction.java** files on-line.
- In comments, the assignment Header described in "Important Note".
- The files must compile and run as you submit it. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission. **NO LATE ASSIGNMENTS WILL BE ACCEPTED.**