**Assignment 2.2**

1. Write the constructor methods for class **StringInstrument** and **Guitar**.

[Please write your code in intellij IDE then copy-paste only the constructor methods you  
 have written below. This instruction applies for the next questions as well]

**class** MusicInstrument {  
 **private** String **name**= **null**;*// the instrument’s name  
  
 /\*\* Constructor: an instrument with name s \*/* **public** MusicInstrument(String s) {  
 **this**.**name** = s;  
 }  
   
 */\*\* Constructor: an instrument with name "" \*/* **public** MusicInstrument() {  
 **this**.**name**= **""**;  
 }  
}  
  
*/\*\* An instance represents a string instrument \*/***class** StringInstrument **extends** MusicInstrument {  
 **private int numStrings**; */\*\* number of strings on this instrument \*/  
  
 /\*\*Constructor: a string instrument with name s and n numStrings\*/* **public** StringInstrument(String s, **int** n) {  
   
 }  
  
 */\*\* Constructor: a string instrument with name "" and -1 numStrings\*/* **public** StringInstrument() {  
   
 }  
   
}  
  
**class** Guitar **extends** StringInstrument{  
 **private** String **type**; */\*type of this instrument\*/  
  
 /\*\*Constructor: a guitar instrument with name s and n numStrings and type p\*/* **public** Guitar(String s, **int** n, String p) {  
  
 }  
  
 */\*\*Constructor: a guitar instrument with name "" and -1 numStrings and type ""\*/* **public** Guitar() {  
  
 }  
}

1. A *superellipse* is a curve whose points satisfy the equation:

An *ellipse* is a curve whose points satisfy the equation:

A *circle* is a curve whose points satisfy the equation:

Complete the three classes: SuperEllipse, Ellipse and Circle given below by completing the bodies of all methods.

**class** SuperEllipse {  
  
 **private double a**;  
 **private double b**;  
 **private double n**;  
  
 *//Precondition: a > 0, b > 0, n > 0* **public** SuperEllipse(**double** a, **double** b, **double** n) {  
 }  
 **public double** getA() {  
 }  
 **public double** getB() {  
 }  
 **public double** getN() {  
 }  
}  
  
**class** Ellipse **extends** SuperEllipse{  
   
 *//Precondition: a > 0, b > 0* **public** Ellipse(**double** a, **double** b) {  
 }  
   
 */\*\* = area of ellipse: PI \* a \* b \*/* **public double** area() {  
 }  
}  
  
**class** Circle **extends** Ellipse {  
   
 *//Precondition: r > 0* **public** Circle(**double** r) {  
 }  
 **public double** getRadius() {  
 }  
}

1. Implement the missing methods of class Positive and Rational.

You do not need to implement method **public void** reduce() of class Rational.

But bonus marks will be provided if anyone does so.

**class** Positive{  
  
 **private int k**; *// the positive integer  
 /\*\* Constructor: an instance with value k.  
 Precondition: k > 0 \*/* **public** Positive (**int** k) {  
 }  
 */\*\* = this instance’s value \*/* **public int** getPositive() {  
 }  
 */\*\* Set the value of this instance to n.  
 Precondition: n > 0 \*/* **public void** setPositive(**int** n){  
 }  
}  
  
  
*/\*\* An instance is a rational number \*/***class** Rational **extends** Positive {  
 */\*\* The rational number is num / k, where k is the value  
 wrapped by the super class.  
 Restrictions on fields: k is always > 0 and  
  
 num / k is always in lowest possible terms.  
 E.g. instead of 10/5 or –5/10, these numbers  
 are stored as 2/1 and –1/2.  
 private int num;  
  
 /\*\* Constructor: an instance with rational number num / denom.  
 Precondition: denom != 0 \*/* **public** Rational(**int** num, **int** denom) {  
 }  
 */\*\* Set the value of the denominator to n.  
 Precondition: n > 0 \*/* **public void** setPositive(**int** n){  
 }  
  
 */\*\* Reduce this rational number to the lowest  
 possible terms, e.g. 8/24 becomes 1/3 \*/* **public void** reduce(){  
  
 }  
}