Lab 1: ADT Pair

```
Location: CourseWeb -> Labs/Recitations -> Lab 1: Pair
Download the following files:
  1. Pair.java
  2. PairInterface.java
  3. PairTester.java
  4. ParabolaFrame.java
  5. GraphComponent.java
```

Introduction

An ordered pair is a simple abstract data type often use in various applications. A pair consists of two objects `x` and `y`. These two objects do not have to be the same type. For example, we can have a pair of string and integer, a pair of floating-point number and string, etc. Generally, a pair of object `x` and `y` is represented by `(x,y)`. Note that `x` or `y` can also be a pair (e.g., `(x,(y,z))`). The object `x` is called the first element and the object `y` is called the second element. Two pairs `(x,y)` and `(a,b)` are said to be equal if and only if `x` is equal to `a` and `y` is equal to `b`. The order of a pair can be reversed by simply swap the first and the second elements. For example, the reverse order of the pair `(x,y)` is the pair `(y,x)`. The behavior of the ADT pair can be represented as a Java interface named `PairInterface` as shown below:

```
public interface PairInterface<T1,T2>
{
        /**
         * Gets the first element of this pair.
         * @return the first element of this pair.
         */
        public T1 fst();

        /**
         * Gets the second element of this pair.
         * @return the second element of this pair.
         */
        public T2 snd();

        /**
         * Sets the first element to aFirst.
         * @param aFirst  the new first element
         */
        public void setFst(T1 aFirst);

        /**
         * Sets the second element to aSecond.
         * @param aSecond  the new second element
         */
        public void setSnd(T2 aSecond);
```

```
        /**
         * Checks whether two pairs are equal. Note that the pair
         * (a,b) is equal to the pair (x,y) if and only if a is
         * equal to x and b is equal to y.
         *
         * Note that if you forget to implement this method, your
         * compiler will not complain since your class inherits this
         * method from the class Object.
         *
         * @param otherObject the object to be compared to this object.
         * @return true if this pair is equal to aPair. Otherwise
         * return false.
         */
        public boolean equals(Object otherObject);

        /**
         * Generates a string representing this pair. Note that
         * the String representing the pair (x,y) is "(x,y)". There
         * is no whitespace unless x or y or both contain whitespace
         * themselves.
         *
         * Note that if you forget to implement this method, your
         * compiler will not complain since your class inherits this
         * method from the class Object.
         *
         * @return a string representing this pair.
         */
        public String toString();
}
```

What to do

You are going to implement the abstract data type Pair by creating a **generic class** named `Pair` that implements `PairInterface`. For this lab, what you have to do is to modify the starting code (`Pair.java`). You need to add instance variables, implement the constructor, and implement all methods that marked `TO DO`. Note that it is important for you to implement the method `toString()` correctly. This method will be used to test your class. If you incorrectly implement the method `toString()` the test result will be incorrect. For example, a pair of string and class wrapper integer containing string `Hello` and integer 5, the method `toString()` should return the string `(Hello,5)`.

Test Class

A test class (`PairTester.java`) is provided. Simply compile and run this test class. This program will test your `Pair.java` and show your total points (out of 10). If you do not get the full 10 points, you should keep trying to fix your code until you get 10 points. **Note** that this test class is not perfect. It cannot tell you why your program is incorrect. You may have to look at the source

code of `PairTester.java` and see why it says `FAIL` and trace your code.

Example of Using the ADT Pair

An example of how the ADT Pair is used is a series of coordinates. Imagine if you have to write a program to plot the parabola graph of the following equation:

$$y = \frac{8}{25}x^2 - 3$$

You need a series of values `(x,y)` and use them to plot the graph by converting each pair of value to a coordinate. To generate a series of values of x from -5.0 to 5.0 (0.01 increment) of the above equations we need to do the following:

1. Create an array list of Pair

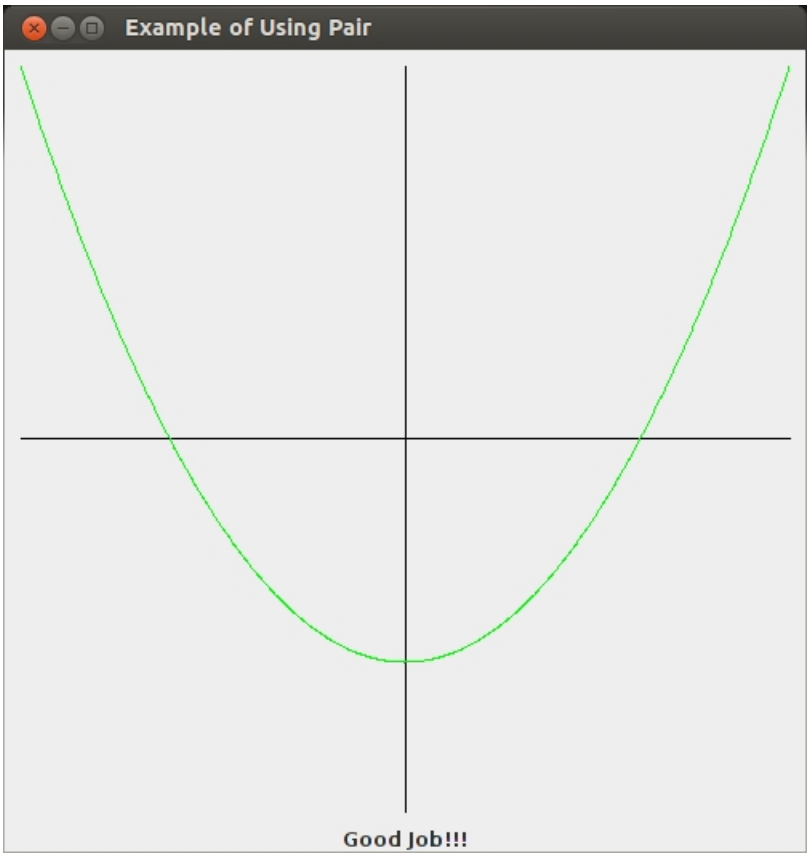
```
ArrayList<Pair<Double,Double>> data = new ArrayList<Pair<Double,Double>>();
```

2. Generate series of values using the above equation and add them to the above array list:

```
for(double i = -5.0; i <= 5.0; i = i + 0.01)
{
        data.add(new Pair<Double,Double>(i,(((i*i)*8)/25)-3));
}
```

3. Use the above array list to draw the graph.

The program `ParabolaFrame.java` is an example of a program that uses the ADT Pair to store a value, generate a series of values, and draw the graph. This program will use **your** `Pair.java`. So, make sure that your implementation works correctly. The output of the program is shown below:



Due Date and Submission

For the due date, please check the lab in the CourseWeb. Submit your `Pair.java` to the CourseWeb under this lab by the due date. **No late submission will be accepted**.