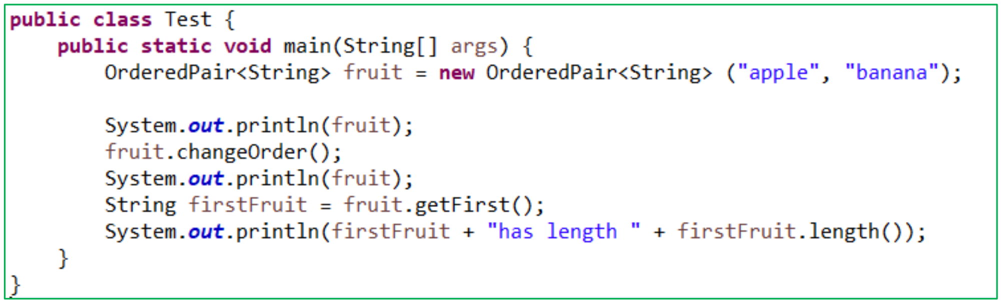
Programming Exercises

```
/**
   An interface for pairs of objects.
*/
public interface Pairable<T>
{
   public T getFirst();
   public T getSecond();
   public void changeOrder();
} // end Pairable
```

Pairable.java

Test.java



```
public class Test {
    public static void main(String[] args) {
        OrderedPair<String> fruit = new OrderedPair<String> ("apple", "banana");

        System.out.println(fruit);
        fruit.changeOrder();
        System.out.println(fruit);
        String firstFruit = fruit.getFirst();
        System.out.println(firstFruit + "has length " + firstFruit.length());
    }
}
```

(apple, banana)
(banana, apple)
Banana has length 6

```
/** A class of ordered pairs of objects having the same data type. */
public class OrderedPair<T> implements Pairable<T>
{
   private T first, second;

   public OrderedPair(T firstItem, T secondItem)
         // NOTE: no <T> after constructor name
   {
      first = firstItem;
      second = secondItem;
   } // end constructor

   /** Returns the first object in this pair. */
   public T getFirst()
   {
      return first;
   } // end getFirst
   /** Returns the second object in this pair. */
   public T getSecond()
   {
      return second;
   } // end getSecond

   /** Returns a string representation of this pair. */
   public String toString()
   {
      return "(" + first + ", " + second + ")";
   } // end toString

   /** Interchanges the objects in this pair. */
   public void changeOrder()
   {
      T temp = first;
      first = second;
      second = temp;
   } // changeOrder
} // end OrderedPair
```

OrderedPair.java