# Stock class

The following program implements a **Stock** class and a short test program for it. Read through the code and then answer the questions on the next page.

Stock.java

```java
class Stock {
  private String symbol;
  private String name;
  private double previousClosingPrice;
  private double currentPrice;

  public Stock() {
  }

  public Stock(String newSymbol, String newName) {
    symbol = newSymbol;
    name = newName;
  }

  public double getChangePercent() {
    return (currentPrice - previousClosingPrice) /
      previousClosingPrice;
  }

  public double getPreviousClosingPrice() {
    return previousClosingPrice;
  }

  public double getCurrentPrice() {
    return currentPrice;
  }

  public void setCurrentPrice(double newCurrentPrice) {
    currentPrice = newCurrentPrice;
  }

  public void setPreviousClosingPrice(
                    double newPreviousClosingPrice) {
    previousClosingPrice = newPreviousClosingPrice;
  }
}
```

TestStock.java

```java
public class TestStock{
  public static void main(String[] args) {
    Stock stock = new Stock(
                     "SUNW", "Sun MicroSystems Inc.");
    stock.setPreviousClosingPrice(100);

    // Set current price
    stock.setCurrentPrice(90);

    // Display stock info
    System.out.println("Previous Closing Price: " +
      stock.getPreviousClosingPrice());
    System.out.println("Current Price: " +
      stock.getCurrentPrice());
    System.out.println("Price Change: " +
      stock.getChangePercent() * 100 + "%");
  }
}
```

## Questions:

1. How do the two files work together as a single program? Which method of which class is called first?

2. Name all of the data fields in the `Stock` class? What do they represent? What does the `private` specifier imply?

3. In how many ways can a `Stock` object be constructed? Which are the methods that are used for constructing `Stock` objects?

4. What are the methods `get...` used for? What is the name of such methods in general?

5. What are the methods `set...` used for? What is the name of such methods in general?

6. What is the purpose of the `TestStock` class?

7. Are the tests in the `TestStock` class sufficient to tell that the `Stock` class is implemented correctly? If no, come up with a few statements that would ”break” the program.

8. What changes should be made to both classes to make the program better:

   - for the `Stock` class, we want to make it resilient to errors that can be cause by misuse of the code by the client code;
   - for the `TestStock` class, we want to make sure that it really tries to break the code of the `Stock` class and that it test if the class behaves correctly.