```
/* Project 1 - Harley Phung
   Task 1: Create Stock class that contained company name, ticker symbol, current
price of the stock, number of shares owned, capital gains.
   That can calculate when there's transaction such as buy, sell, or split
 */

public class Stock {
  //A field that stored company's name
  private String companyName = " ";
  //A field that stored market's Ticker symbol
  private String tickerSymbol = " ";
  //A field that stored the current currentPrice of stock
  private double currentPrice = 0;
  //A field that stored number of shares of Stock
  private int numShares = 0;
  //A field that stored the cost basis
  private double costBasis = 0;
  //A field that stored capital gains
  private double capitalGains = 0.0;
  //A field that stored commission
  private double commission = 0;

  //Constructor 1, contain ticker symbol and stock's current price
  public Stock(String tickerSymbol, double currentPrice) {
    this.tickerSymbol = tickerSymbol;
    this.currentPrice = currentPrice;
  }

  //Constructor 2 contains company name, ticker symbol. and stock's current price
  public Stock(String companyName, String tickerSymbol, double currentPrice) {
    this.companyName = companyName;
    this.tickerSymbol = tickerSymbol;
    this.currentPrice = currentPrice;
  }

  // getCompanyName return company name
  public String getCompanyName(){
    return this.companyName;
  }

  // setCompanyName changes the company name
  public void setCompanyName(String companyName) {
    this.companyName = companyName;
  }

  //getTickerSympol returns the market's ticker symbol
  public String getTickerSymbol() {
    return this.tickerSymbol;
  }

  //setTickerSymbol changes the market's ticker symbol
  public void setTickerSymbol(String tickerSymbol) {
    this.tickerSymbol = tickerSymbol;
  }

  //getCurrentPrice return the current price for a share of the stock
  public double getCurrentPrice() {
    return this.currentPrice;
  }
```

```
  //setCurrentPrice changes the current price of a share of the stock
  public void setCurrentPrice(double currentPrice) {
    this.currentPrice = currentPrice;
  }

  //getNumberShares returns the number of shares owned in stock
  public int getNumberShares() {
    return this.numShares;
  }

  //getCostBasis returns the cost basis for the owned shares
  public double getCostBasis() {
    return this.costBasis;
  }

  //getCapitalGains reutrns the total cpital gains earned so far
  public double getCapitalGains() {
    return this.capitalGains;
  }

//A method that will increase the number of owned number of shares to an amount and
change the cost basis.
  public double buy(int numShares, double commission) {
    this.numShares = this.getNumberShares() + numShares;
    this.costBasis = this.getCostBasis() + numShares * this.getCurrentPrice() +
commission;
    return numShares * this.getCurrentPrice() + commission;
  }

  //A method that sell the number of owned shares and changes number of shares,
capital gains, and cost basis
  public double sell(int numShares, double commission) {
    if (numShares > this.getNumberShares()){
      return 0;
    }

    else{
      this.capitalGains = this.getCapitalGains() + ((numShares *
this.getCurrentPrice() - commission) - (this.costBasis * ((double)numShares /
(double)this.getNumberShares())));
      this.costBasis = this.costBasis - this.costBasis *
(double)numShares/(double)this.getNumberShares();
      this.numShares = this.getNumberShares() - numShares;
      return (numShares * this.getCurrentPrice() - commission);
    }
  }

  //A method that check if the stock account's remainder need to be sold
  public double split(int ratioNumerator, int ratioDenonimator) {
    if(ratioNumerator <= 0 || ratioDenonimator <= 0){
      return 0;
    }

    else {
      double charges = (double) ratioNumerator / (double) ratioDenonimator; //
charges represents the ratio of ratioNumerator over ratioDemonimator
      double tempoNumShares = (double)this.numShares * charges; // tempoNumShares
represent the number of Shares when multiplies the charges.
```

```
      if (tempoNumShares % (int)tempoNumShares == 0) {
        this.numShares = (int)tempoNumShares;
        return 0;
      }
      else {
        double fractionalShare = tempoNumShares - (int)tempoNumShares; //the
remainder when subtracted the double to int. For example, 2.5-2 = 0.5, the 0.5 is
fractionalShare
        this.capitalGains = this.getCapitalGains() + (fractionalShare *
this.getCurrentPrice() - (this.costBasis * (fractionalShare / tempoNumShares)));
        this.costBasis = this.costBasis - (this.costBasis * fractionalShare /
tempoNumShares);
        this.numShares = (int)(tempoNumShares - fractionalShare);
        return fractionalShare * this.getCurrentPrice();
      }
    }
  }
}
```