```java
import java.io.*;
import java.util.*;
import java.util.jar.JarOutputStream;

public class SalesSplit
{
   /**
      Prompts for and reads name of file to process.
      @param in Scanner from which to read
      @return file name
   */
   private static String getFileName(Scanner in)
   {
      String filename = null;
      System.out.print("Enter file to process: ");
      if (in.hasNext())
      {
         filename = in.next();
      }

      return filename;
   }

   /**
      Read a sales entry from input Scanner.
      @param in Scanner from which to read entry
   */
   private static Sale readSaleEntry(Scanner in)
   {
      String inputLine = in.nextLine();
      Scanner lineScan = new Scanner(inputLine);
      lineScan.useDelimiter(";");
//YOUR CODE TO INPUT name, service, amount, and date.
      int i = 0;
      String name = ""; String service=""; double amount=0; String date="";
      while(lineScan.hasNext()){
         String value=lineScan.next();
         if(i%4==0) {
            name=value;
         }
         else if(i%4==1) {
            service=value;
         }
         else if(i%4==2) {
            amount=Double.parseDouble(value);
         }
         else if(i%4==3) {
            date=value;
         }
         i++;
      }
      Sale return_sale = new Sale(name, service, amount, date);
      return return_sale;
   }
```

```
  /**
    Reads sales data from the specified file.
    @param filename name of sales data file
    */
   private static ArrayList<Sale> readSalesFile(String filename) throws FileNotFoundException
   {
      ArrayList<Sale> sales = new ArrayList<Sale>();

      if (filename != null)
      {
         try (Scanner infile = new Scanner(new File(filename)))
         {
            while (infile.hasNext()){
               sales.add(readSaleEntry(infile));
            }
         }
         catch(IllegalArgumentException e)
         {
            System.out.println("File format not valid.");
         }
      }
      return sales;
   }


   /**
    Writes sale to writer.
    @param out PrintWriter to which sale is written
    @param sale sale information
    */
   private static void writeSale(PrintWriter out, Sale sale)
   {
      out.print(sale.getName());
      out.print(";");
      out.print(sale.getServiceCategory());
      out.print(";");
      out.print(sale.getAmount());
      out.print(";");
      out.print(sale.getDate());
      out.println();
   }

   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      String filename = getFileName(in);
      System.out.println("The filename is: " + filename);

      try
      {
         PrintWriter sales_output = new PrintWriter("salesoutput.txt");
         ArrayList<Sale> sales = readSalesFile(filename);
         for (Sale s : sales) {
```

```
            System.out.println(s);
            writeSale(sales_output, s);
        }
        sales_output.close();


    }
    catch (FileNotFoundException e)
    {
        System.out.println("File not found!");
    }
    catch (NoSuchElementException e)
    {
        System.out.println("File format not valid.");
    }
}
}
```

Sales Output.txt
John Public;Dinner;29.95;6/7/2014
Jane Public;Conference;499.0;8/9/2014
Abby Lawrence;Dinner;23.45;10/10/2014