CS 1410 Introduction to Computer Science – CS2
Section 1: MWF 9:30 a.m. – 10:20 a.m.
Instructor: Xiaojun Qi
Assignment #5

Given: Tuesday, Feb. 12, 2013
Due: 11:59 p.m. Thursday, Feb. 21, 2013
Total Points: 30 points

Package delivery services such as FedEx®, DHL®, and UPS® offer a number of different shipping options, each with specific associated costs. For this programming assignment, you are going to create an inheritance hierarchical structure that allows the user to calculate the shipping cost for different shipping options or packages.

To solve this problem, you will need to create three classes: the base class **Package**, and two derived classes **TwoDayPackage** and **OvernightPackage**. In addition, you will also need to create a separate class called **Person** which contains five private data members. Each of these data members is a string denoting name, address, city, state, and zip-code. **Person** should have two constructors, one with five string parameters that initializes the five strings in the class and one with no parameters that initializes everything to empty.

Each **Package** object has two **Person** objects, one for the sender and one for the receiver. Also, a **Package** will contain three private attributes, i.e. weight (in ounces), cost per ounce to ship, and number of packages shipped to date. The **TwoDayPackage** and **OvernightPackage** each inherit or derive all of the attributes and methods of a **Package**. Additionally, an **OvernightPackage** will include an attribute for a flat rate increase to the cost of shipping the package, and an attribute for the number of **OvernightPackages** shipped to date. A **TwoDayPackage** will include an increase to the regular **Package** cost per ounce rate, and an attribute for the number of **TwoDayPackages** shipped.

In summary, the attributes of these three classes are:

Package
- Sender Person Object
- Receiver Person Object
- Weight in ounces
- Cost per ounce
- Number of Packages Shipped (static)

OvernightPackage
- Inherits all attributes of Package
- Flat rate increase
- Number of Overnight Packages Shipped (static)

TwoDayPackage
- Inherits all attributes of Package
- Cost per ounce increase

Number of TwoDayPackages Shipped (static)

Your program should read from a user specified **"package.txt"** file and validate that the file is present in the current working directory. **You may assume that this file is correctly formatted as follows**:

(Note: Package uses lines 1-13, TwoDayPackage and OvernightPackage use lines 1-14)

Line 1: <package type>	// P, O, or T (not case sensitive) denoting package type
Line 2:<integer>	// weight in ounces
Line 3:<double>	// cost per ounce, e.g. 7.512 => $7.512 per ounce
Line 4:<string>	// Sender name
Line 5:<string>	// Sender address
Line 6:<string>	// Sender city
Line 7:<string>	// Sender state
Line 8:<string>	// Sender zip code (5 digits)
Line 9:<string>	// Receiver name
Line 10:<string>	// Receiver address
Line 11:<string>	// Receiver city
Line 12:<string>	// Receiver state
Line 13:<string>	// Receiver zip code (5 digits)

TwoDayPackage (one additional line)

Line 14:<double>	// Amount to add to the per ounce cost for a regular package

OvernightPackage (one additional line)

Line 14:<double>	// the fixed amount to add to the calculated cost for a one day
	// package

Please create your own "package.txt" file for this assignment and refer to page 131 on the textbook for details on how to perform file operations.

As an example, consider a package with the following attributes.

Package weight:	72 ounces
Package cost per ounce:	$0.05
OvernightPackage fixed cost	$25.25
TwoDayPackage rate increase	$0.02

Cost to deliver as:

Package	72 * 0.05 = $3.60
OvernightPackage	72 * 0.05 + 25.25 = $28.85
TwoDayPackage	72 * (0.05 + 0.02) = $5.04

After the data for a shipment is read in, your program should instantiate objects of the appropriate class(es), and output a shipping label and the cost to ship. Once you have come to

the end of the file, your program should output the number of packages of the three types that have been received/read.