Assignment 1 – Java and Java OOP

Jing Hua Ye, MTU

Question 1 (Separating the Digits in an Integer)

Write an application that inputs one number consisting of five digits from the user, separates the number into individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in the number 42339, the program should print `4 2 3 3 9`

Question 2 (Bar-Chart Printing Program)

One interesting application of computers is to display graphs and bar charts. Write an application that reads five numbers between 1 and 30. For each number that's read, your program should display the same number of adjacent asterisks. For example, if your program reads the number 7, it should display *******. Display the bars of asterisks after you read all five numbers.

Question 3 (Airline Reservations System)

A small airline has just purchased a computer for its new automated reservations system. You've been asked to develop the new system. You're to write an application to assign seats on each flight of the airline's only plane (capacity: 10 seats).

Your application should display the following alternatives: Please type 1 for First Class and Please type 2 for Economy. If the user types 1, your application should assign a seat in the first-class section (seats 1 - 5). If the user types 2, your application should assign a seat in the economy section (seats 6 - 10). Your application should then display a boarding pass indicating the person's seat number and whether it's in the first-class or economy section of the plane.

Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it's acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours".

Question 4 (Invoice Class)

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables - a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount, then return the amount as a double value. If the quantity is not positive,

it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test app named InvoiceTest that demonstrates class Invoice's capabilities.

Question 5 (Book Hierarchy)

Books can come in various formats, like paper books, audio books, ebooks, etc. Create a generic class Book that has as common attributes the title, the year of publication, and the author. The constructor of this class should instantiate all three attributes. Override the toString method of class Book that returns a string that contains the values of its attributes. Create a subclass PrintBook that extends Book with attributes Publisher and ISBN. Create another subclass AudioBook which has the book's size (in MB), its play length and the playback artist's name as attributes. Both PrintBook and AudioBook classes override the toString method inherited from Book. Write a Java application to demonstrate the usage of this hierarchy.