

Object-Oriented Programming in Java

MISM/MSIT 95-712

Project 1

This first homework is intended to help you get comfortable with the edit-compile-run cycle of Java programming. If you've had previous experience programming, writing the code itself should not be difficult. If writing the actual code (as opposed to learning to use your programming tools) takes longer than an evening, you should consider talking with the TAs or me.

1. (25 points) Write a program that inputs the name, quantity, and price of three items. The name may contain spaces. Output a bill with a tax rate of 5.25%. All prices should be output to two decimal places. The bill should be formatted in columns with 30 characters for the name, 10 characters for the quantity, 10 characters for the price, and 10 characters for the total. Sample input and output is shown as follows:

Input name of item 1:

Freddo

Input quantity of item 1:

10

Input price of item 1:

0.50

Input name of item 2:

Coke

Input quantity of item 2:

3

Input price of item 2:

1.25

Input name of item 3:

Mini Chips

Input quantity of item 3:

20

Input price of item 3:

0.75

Your bill:

Item	Quantity	Price	Total
lollipops	10	0.50	5.00
diet soda	3	1.25	3.75
chocolate bar	20	0.75	15.00
Subtotal			23.75
6.25% sales tax			1.48
Total			25.23

2. (25 points) Use the `System` or `Calendar` class from the JAVA class library to test how fast you can type the sentence “**I type very quickly**”. Your program should print:

```
Your job is to type the sentence "I type very quickly" as fast as you
can.
When you are ready, press enter, type the sentence, and press enter
again.

Now press enter...
```

Your program should print the number of milliseconds the user took to type the sentence. Of course you must check to see if the user typed the sentence correctly, and print a message if the user input is incorrect. Correct case is important (use the `String` class). My personal best time is 6325 milliseconds. See if you can beat me!

3. (50 points) Read the following case study, which describes the data requirements for a video rental company. The video rental company has several branches throughout the USA. The data held on each branch is the branch address made up of street, city, state, and zip code, and the telephone number. Each branch is given a branch number, which is unique throughout the company. Each branch is allocated staff, which includes a Manager. The Manager is responsible for the day-to-day running of a given branch. The data held on a member of staff is his or her name, position, and salary. Each member of staff is given a staff number, which is unique throughout the company. Each branch has a stock of videos. The data held on a video is the catalog number, video number, title, category, daily rental, cost, status, and the names of the main actors, and the director. The catalog number uniquely identifies each video. However, in most cases, there are several copies of each video at a branch, and the individual copies are identified using the video number. A video is given a category such as Action, Adult, Children, Drama, Horror, or Sci-Fi. The status indicates whether a specific copy of a video is available for rent. Before hiring a video from the company, a customer must first register as a member of a local branch. The data held on a member is the first and last name, address, and the date that the member registered at a branch. Each member is given a member number, which is unique throughout all branches of the company. Once registered, a member is free to rent videos, up to maximum of ten at any one time. The data held on each video rented is the rental number, the full name and number of the member, the video number, title, and daily rental, and the date the video is rented out and date returned. The rental number is unique throughout the company.

- (a) Write a problem scope statement (1 paragraph) for the above business scenario.
- (b) Identify the main objects for the system you have identified for the above business.
- (c) Identify the attributes and behaviours for the objects in the objects and create a table where you will list the high-level classes that are included in this problem. You can create a table (like our lecture example) with objects including the attributes and operations for each.

Your final submission will include *a pdf file with solutions to Question 1 plus a JAVA source file each containing the solution to Questions 2 and 3*. Before you make your submission, put **all** the files into a **single** compressed (zip) folder, and submit only this **one** file. Be sure to choose names for all your files that reflect their content. The name of the zip file you submit should contain your own name, and the assignment number, for example, `MurliViswanathanProj1.zip`.

For every homework assignment, Canvas will allow only one file to be submitted. This must be the zip file discussed above. Also, once you have submitted a file to Canvas, you are not allowed to remove it and re-submit another file. Because of this, you should be *absolutely sure* that you are finished with the homework and that it is correct before you submit it.