

# CS1073 - Assignment #5 - Winter 2024

---

**Submission Deadline: Thursday, March 14th before 12:00 NOON (Atlantic) in the Assignment 5 submission folder in Desire2Learn. (Read the submission instructions at the end of this document carefully).**

The purpose of this assignment is to review/gain practice with:

- to review earlier coverage of objects and classes,
- to introduce the "is a" relationship (inheritance),
- decision statements,
- the `while` loop,
- arithmetic in Java.

This assignment is to be done individually. What you hand in must be your own work. Incidents of plagiarism will be reported.

If you have questions about the assignment, you should first go to a scheduled help session. (Locations and times for all help sessions can be found on D2L). If you have attended a help session and the issue is unresolved, you may contact your course instructor. You are NOT to discuss this assignment with anyone else (including your classmates).

---

As always, begin by creating a new folder to hold your work for this assignment.

## I. Ticket App

### A. Basic Inheritance:

Write a class named **TrainTicket** that can be used to represent a train ticket for a trip. Every train ticket has the name of the passenger, the number of checked bags, and the trip distance, and we must be able to retrieve each of these pieces of information for any ticket. (Note: train tickets are non-transferrable and the trip cannot be changed – this means the name of the passenger and the trip distance can never change once the ticket is created.) We must also be able to compute the cost of the train when told the per km rate for the ticket using the method: **computeCost(double rate)**. The cost of the ticket must also include the cost of checked bags which is \$45 per bag.

For example, if the passenger has 3 checked bags, the trip distance is 1050km and the per km rate is \$1.50, the computeCost method will return 1635.

Next, create a subclass of **TrainTicket** named **BusinessTicket**. This class will be used to represent a train ticket that is in the business train carriages which have larger seats and more space. The cost of a business ticket has an additional charge of \$104.50 added to the base cost of a train ticket. The business train cars have both regular seats and reclining seats. Customers may choose to have a reclining seat for an additional \$55.75 charge.

Finally, create a third class named **SleeperCarTicket**. This class will also be a subclass of **TrainTicket**. Sleeper car tickets are for carriages that contain sleepers – these are seats can be converted into beds. Sleeper car tickets must also keep track of the number of nights the passenger wants to reserve the sleeper for. There is an additional charge of \$167 added to the base cost for a sleeper car ticket. In addition to this, there is a per night fee. The first night is \$95, and each additional night is \$80.

Be sure to write javadoc comments for all three classes. Include a comment for the class, for each instance variable and constant, and for each method. Use @author, @param & @return tags where appropriate.

## B. Test Driver

Write a test driver named **TicketApp** that exercises many of the methods from the TrainTicket, BusinessTicket and SleeperTicket classes. Use the following test values:

Passenger Name	Trip Distance (in km)	Number of checked bags	Has Reclining seat	Number of nights
Grace Hopper	1150	2	Not applicable	Not applicable
Joy Buolamwini	820	1	Not applicable	Not applicable
Tim Berners-Lee	755	2	Not applicable	Not applicable
Margaret Hamilton	690	0	Yes	Not applicable
Barbara Liskov	1402	3	No	Not applicable
Shafi Goldwasser	975	1	Yes	Not applicable
Alan Turing	3327	4	Not applicable	3

Lynn Conway	1766	0	Not applicable	1
Jeannette Wing	5003	3	Not applicable	4

- Grace purchased her ticket during peak season at a per km rate of \$1.35/km.
- Joy purchased her ticket using her frequent rider card and received a reduced rate of \$0.75/km.
- Tim purchased his ticket during the off season at a per km rate of \$1.15/km.
- Margaret purchased her ticket during peak season also, her rate was \$1.20/km.
- Barbara purchased her ticket during the off season at a per km rate of \$0.95/km.
- Shafi and Alan both purchased their tickets during a sale at a per km rate of \$0.85/km.
- Lynn and Jeanette both purchased their tickets during a special promotion at a per km rate of \$0.65/km.

Your program will produce output in a table, *formatted as shown here* (this table does not contain all the information – it is only a sample):

Passenger & Trip Distance	Bags	Cost
=====	=====	=====
Grace Hopper (1150.0km)	2	\$1,642.50
Joy Buolamwini (820.0km)	1	\$660.00
Tim Berners-Lee (755.0km)	2	\$958.25

(Note: In your driver, use the NumberFormat class to format the currency values when producing output.)

Be sure to write a javadoc comment for your TicketApp class. This comment should include a one-line description of the class and @author information.

After you have tested your application and you're sure that it works properly, save the sample output to a text file (using re-direction) to include in your archive. For your lab report, you may copy and paste the output from the terminal window into your report.

Continued on next page...

## II. Arabic to Ancient Number System:

An ancient numbering system was recently discovered that used a base 16 positional numeral system. The system is made up of three symbols: I (vertical bar) to represent 1, M to represent 4 and X to represent zero. The first row represents  $16^0$ , the next row represents  $16^1$ , the next is  $16^2$  and so on (the highest power is in the last row).

Symbol	Value
	1
M	4
x	0

The symbols are drawn with the smallest place value ( $16^0$ ) on the top row, and each larger place value written below. For example 729 would be written:

MM|            =  $9 \times 16^0$   
MMM|         =  $13 \times 16^1$   
| |            =  $2 \times 16^2$

Write a program that prompts the user for a positive integer and then prints out the number in ancient numerals. The program can be written inside of the main method of a class.

**Sample Output** (note: user input is shown in bold and italics):

```
Enter a positive integer: 729
The ancient number for 729 is:
MM|
MMM|
| |
```

**Sample Output** (note: user input is shown in bold and italics):

```
Enter a positive integer: 4096
The ancient number for 4096 is:
X
X
X
|
```

Continued on next page...

**Sample Output** (note: user input is shown in bold and italics):

```
Enter a positive integer: 271  
The ancient number for 271 is:  
MMM|||  
X  
|
```

After you have tested your application (with several different input values) and you're sure that it works properly, save sample output from running the program 3 times (3 different test cases). For this question, take a snapshot of the terminal window to include in your .zip file. That way, the marker will be able to see the input values as well. For your assignment report, you may copy and paste the output from the terminal window into your report (again, this will ensure that the input values are also included.)

**Submission instructions are on the next 2 pages...**

**Your electronic assignment submission (submitted via Desire2Learn) will consist of two files:**

- i. a written report. This should begin with a title page; your title page should include: the course (CS 1073), your section (FR01A, FR02A, FR03A, FR04A, FR05A or FR06A), the assignment number (Assignment #5 in this case), your full name, and your UNB student number. That should be followed by the sections below, with each part clearly identified with a section heading. Include:
  - a. the source code for Part I,
  - b. the sample output for Part I,
  - c. the source code for Part II, and
  - d. the sample output for Part II.

(Aside: Your source code should contain all of the javadoc comments mentioned above. However, you do not need to include the .html files in your report.)

This written report should be prepared using a word processor. (Options were outlined in Lab #1. If you choose to use the online version of MS Word, please see the note from Lab #1 about the issue when copying and pasting tabs into a document.)

Copy & paste the items listed above into the report document. (These should appear in the document in the order that they are listed above). Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code and question II output to maintain proper indentation. (Examples of monospaced fonts were mentioned in Lab #1.)

**Continued on the next page...**

Once the report is complete and you've checked it all over, save the final document file for your own records. Then **save a second copy in PDF format for submission**. (Note: Be sure to open the second file in a PDF viewer to verify that the PDF was generated correctly.) The **single .pdf file** containing your report will be submitted to the appropriate submission folder in Desire2Learn. (It is important that you submit a PDF file and NOT the original Word or LibreOffice document. This PDF will allow the marker to write comments directly on your work to give you better feedback.)

Note: Please name this report as follows: **YourName\_As5\_Report.pdf**

- i. an archive file (.zip) that contains your Java source code and output for this assignment. Make sure that your archive includes **all the .java files** (in case the marker wishes to compile & run your code to test it), the output from testing your code. You should not include the report document or the .class files in your archive. This archive should be submitted as a single .zip file to the appropriate submission folder on Desire2Learn.

Side note: While we encourage you to run the javadoc tool on your .java files (to check to make sure that you wrote the Javadoc comments correctly), you do not need to add to your archive all of the files & folders that the javadoc utility creates.

This archive should be submitted as a **single .zip file** to the appropriate submission folder in Desire2Learn.

Note: Please name this archive file as follows: **YourName\_As5\_Archive.zip**

**Reminder: Your submission in Desire2Learn should consist of TWO files (a .pdf and a .zip). Do NOT put your report inside your archive.**

---

**End of Assignment 5**