GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Department of Computing & Mathematics

B.Sc. Software Development – Advanced Object-Oriented Design Principles & Patterns (2013)
ASSIGNMENT DESCRIPTION & SCHEDULE
A String Encoding & Decoding API

Note: This assignment will constitute 50% of the total marks for this module.

1. Overview

String encoding is the process of converting a sequence of characters (bytes) in a string from one form to another. We have already seen in Distributed Systems, how encoding can be used to marshal data before transmission over a network. While encoding is implicit in all strings, through the underlying use of ASCII/Unicode standards to map characters to numbers, encoding can also be applied as a prelude to compression and encryption. Commonly use string encoding techniques include Huffman encoding, Lempel–Ziv (LZ77 and LZ78), Run-length and Base64 encoding. You are required to develop a Java Encoding & Decoding API that can parse a file or a URL and encode the data using a variety of different algorithms. Your API should consist of a loosely-coupled design of highly cohesive classes that allow a programmer to quickly and easily exploit the underlying functionality exposed by your library. In addition to designing and implementing the API, you are also required to unit test your library using JUnit and provide an Ant build script with your submission.

Note that the whole point of this assignment is for you to demonstrate an understanding of the principles of object-oriented design by using abstraction, encapsulation, composition, inheritance and polymorphism WELL throughout the application. Hence, you are also required to provide a UML diagram of your design and to JavaDoc your code. Please pay particular attention to how your application must be packaged and submitted. Marks will be deducted if you deviate (even slightly…) from the requirements. Finally, as 4th year software students, you should appreciate that, if your code does not compile you cannot pass the assignment…

2. Deployment and Submission

- ***The project must be submitted by midnight on Sunday 19th January 2014.*** The project must be submitted as a Zip archive ***(not a rar or WinRar file)*** using the Moodle upload utility. You can find the area to upload the project under the "String Encoding & Decoding API (50%) Assignment Upload" heading in the "Notices and Assignments" section of Moodle.
- The name of the Zip archive should be <*id*>.zip where <*id*> is your GMIT student number.
- The Zip archive should have the following structure (do NOT submit the assignment as an Eclipse project):

Component	Category
src	A directory that contains the packaged source code for your application.
README.txt	A text file outlining the main features of your application, with a set of instructions for running the programme.
design.gif	A UML diagram of your API design. Your UML diagram should only show the relationships between the key classes in your design. Do not show methods or variables in your class diagram.
docs	A directory containing the JavaDocs for your application.
build.xml	An Ant build script that compiles the code in the ***src*** directory and builds a JAR archive called ***encoder.jar.*** Use the Ant build script available on Moodle. DO NOT MAKE ANY ALTERATIONS TO THE ANT SCRIPT!

3. Marking Scheme

Marks for the project will be applied using the following criteria:

Marks	Category
(40%)	Robustness
(10%)	Abstraction
(10%)	Encapsulation
(10%)	Unit Tests
(20%)	Packaging & Distribution (including docs and UML)
(10%)	Documented (and relevant) extras.

Each of the categories above will be scored using the following criteria:

- 0–30%: Not delivering on basic expectation
- 40-50%: Meeting basic expectation
- 60–70%: Tending to exceed expectation
- 8-9%: Exceeding expectations
- 10%: Exemplary

Copying & Plagiarism

Please note that the college policy on plagiarism is set out in the Student Code of Conduct (see http://www.gmit.ie/presidents-office/quality-assurance/academic-codes-practice/code-of-practice-no7.pdf). Plagiarism is the passing off of the work of another person as one's own and constitutes a serious infraction that may result in disciplinary proceedings. This assignment is an INDIVIDUAL assignment. While collaboration with other class members is acceptable for high-level design and general problem solving, you must individually code, implement and document your own submission.