DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

CS994 OBJECT ORIENTED PROGRAMMING 2024/25
INDIVIDUAL COURSEWORK ASSIGNMENT

Breakdown of Marking criteria & Checklist of deliverables

The following criteria will be used while marking your submission:

1. Requirements **5 Marks**

1. At least one inheritance relationship/hierarchy: one superclass (parent) with at least two subclasses (children) 2 Marks
2. Documentation of the inheritance relationship/hierarchy requirements 3 Marks

What to submit?

Requirements document (maximum of 300 words). Write a couple of paragraphs describing the inheritance requirements of your system, and the rationale behind your choices. This deliverable is solely about inheritance, not requirements in general; hence, do not copy/paste my final requirements document! Keep this document short. ***300 words is an upper limit, NOT a target!***

2. System design **15 Marks**

Correctness & completeness of class diagram 6 Marks (for inheritance)
9 Marks (for all other classes)

What to submit?

A class diagram.

Please include the inheritance relationships/hierarchy. No need to include unit testing classes in your class diagram. Your design must be consistent with your implementation. Naming conventions apply to system design, too.

3. System implementation (completeness and correctness) **50 Marks**

5 Marks for style

45 Marks for source code – including inheritance (completeness and correctness)

<u>Important note</u>*: Please note that style of source code is part of this marking criterion.*

For style of source code, you are expected to have the following:

- *Consistent (good) style throughout*
- *Code nicely laid-out*
- *Java naming conventions*
- *Code modularisation*
    - *One Java file per class.*
    - *All files in one project.*

What to submit?

All your Java source code, i.e. .java files.

Make sure that you submit all your code, even if it contains syntax errors and/or bugs. Partial marks are available!

4. Testing/Test cases 20 Marks

Correctness & completeness (i.e. fully test the functionality of each class) of **JUnit** tests

Important note: *Please note that commenting and style of source code are part of this marking criterion. The rules for commenting are the same as Marking criterion 5 below, and for style are the same as Marking criterion 3 above. Plus, for style:*

- *One Junit/unit testing class per "main implementation" class.*
- *One Java file per Junit/unit testing class.*

What to submit?

All your Java source code (i.e. .java files) for your JUnit tests.

Make sure that you submit all your JUnit testing code, even if it contains syntax errors and/or bugs. Furthermore, if you produced conventional unit tests instead of JUnit tests, make sure you submit these. Partial marks are available.

5. System documentation/API 10 Marks

Javadoc for all classes and all public methods & constructors, including Junit/unit testing classes (correctness & completeness)

Important note: *Please note that commenting is part of this marking criterion. For commenting, you are expected to have javadoc comments for all classes and all public methods & constructors (correctness & completeness).*

What to submit?

The generated javadoc as .html files.

You can automatically generate the javadoc .html files on BlueJ by switching between "code view" and "documentation view". Ask if in doubt!

Partial marks available.

Important note: *It is important that your API descriptions are detailed in terms of* ***what*** *a class is about or* ***what*** *a method/constructor does, but* ***NOT*** *how it does it (i.e. avoid describing the logic of your implementation). Other programmers should be able to understand your system by only reading its API, i.e. in the same way that we do while reading the Java library API. In order to achieve this, you must include information on: what a class is about, what the method/constructor parameter(s) are, what the method/constructor does (but not how), and what the method outcome is (e.g. what the return value is).*

Good luck!