|  |
| --- |
| cORK iNSTITUTE OF TECHNOLOGY |
| Advanced Object Technology |
| Assignment 1 |
|  |
| **Jimmy Collins – R00145569** |
| **[Pick the date]** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

Contents

[Section 1 – Demonstrated Use of Object Orientated Concepts 1](#_Toc475299507)

[Re-Use 1](#_Toc475299508)

[Encapsulation 1](#_Toc475299509)

[Abstraction 1](#_Toc475299510)

[Inheritance 1](#_Toc475299511)

[Overloading 1](#_Toc475299512)

[Dynamic Binding 1](#_Toc475299513)

[Inner Class 1](#_Toc475299514)

[Nested Class 1](#_Toc475299515)

[Abstract Class 1](#_Toc475299516)

[Generic Method 1](#_Toc475299517)

[Generic Class 1](#_Toc475299518)

[Bounding 1](#_Toc475299519)

[Custom Exception Handler Type 1](#_Toc475299520)

[Section 2 – Usage of Design Patterns 2](#_Toc475299521)

[Strategy Pattern 2](#_Toc475299522)

[Observer Pattern 2](#_Toc475299523)

[Factory Pattern 2](#_Toc475299524)

[Section 3 – System UML Diagrams 2](#_Toc475299525)

[Section 4 – Code Screen Captures 2](#_Toc475299526)

[Section 5 – Evaluation of Work 2](#_Toc475299527)

## Section 1 – Demonstrated Use of Object Orientated Concepts

### Re-Use

Logging?

### Encapsulation

TODO – demonstrated by FlowFileStats.java?

### Abstraction

TODO – ensure to differentiate with Encapsulation

### Inheritance

Illustrated by BinetFileParser extending ParsableFile. More files types could be supported here in the future easily.

### Overloading

Illustrated in my logging class where there are different logging methods (TODO: Move UI logging method from Utility.java to Logger.java, it can be a further overloaded method.

### Dynamic Binding

TODO

### Inner Class

ParsingObserver used in MainController – used to update General tab on the UI as a file parses.

### Nested Class

Used for some listener event?

### Abstract Class

Covered by DisplayStrategy – just make this interface abstract? (Interfaces are inherently abstract).

Covered by ParsableFile.

Covered by Oserver.java

### Generic Method

TODO – used in Utility.java – SortMapByValue

### Generic Class

TODO – ArrayList?

### Bounding

TODO used in Utility.java – GetTopElements

### Custom Exception Handler Type

TODO – for parsing bad info from file?

## Section 2 – Usage of Design Patterns

### Strategy Pattern

This is used to allow choosing of different chart types – pie chart, bar chart etc. Support for other charts could be added easily. User can also change chart types dynamically at runtime.

### Observer Pattern

Used to monitor file parsing info and display on the UI on the general tab as a file parses.

### Factory Pattern

TODO

## Section 3 – System UML Diagrams

TODO – use Visual Paradigm.

## Section 4 – Code Screen Captures

TODO – 1) Generic function.

TODO – 2) File Parsing? Inner class in Main Controller?

## Section 5 – Evaluation of Work