**Iteration 1**

**Requirement**

Store GPS coordinates that define a location and integration with the data created by the user programmer.

**Objectives**

**Create a class that stores normalised GPS coordinates.**

A class containing 2 doubles for latitude and longitude normalised to between -90 – 90 and -180 – 180 which will be required for future calculations. Software reuse was explored in using the GeoCoordinate Class but it was deemed not suitable because of lack of normalisation and it would limit the portability of the class library and the code to create the methods required is widely available.

**Create an abstract class that contains a list of coordinates that can be attached to an existing class**

The ability to integrate with existing data is important to the requires an abstract class that only holds the minimum required(ID, List of points, calculated properties) to use the rest of the library. The class should store an arbitrary list of points that make up a polygon, calculate a bounding box that surrounds it and determine the centre as this will all be required for future calculations.

**Create a generic list that store a list of the location derived from the above abstract class**

The use of Generics will keep the library easy to use and aid in compatibility with the user programmers data classes.

**Completion criteria**

1. The class for storing coordinates should accept any valid double value and normalise latitude to within -90 & 90 and longitude to within -80 & 180.
2. The abstract class should successful recalculate the bounding box and centre as points are added and removed. For performance reasons it should provide the ability to add a range of points and only calculate changes once
3. The class for storing a list of locations should accept different classes derived from the abstract class and allow for added and removing without error