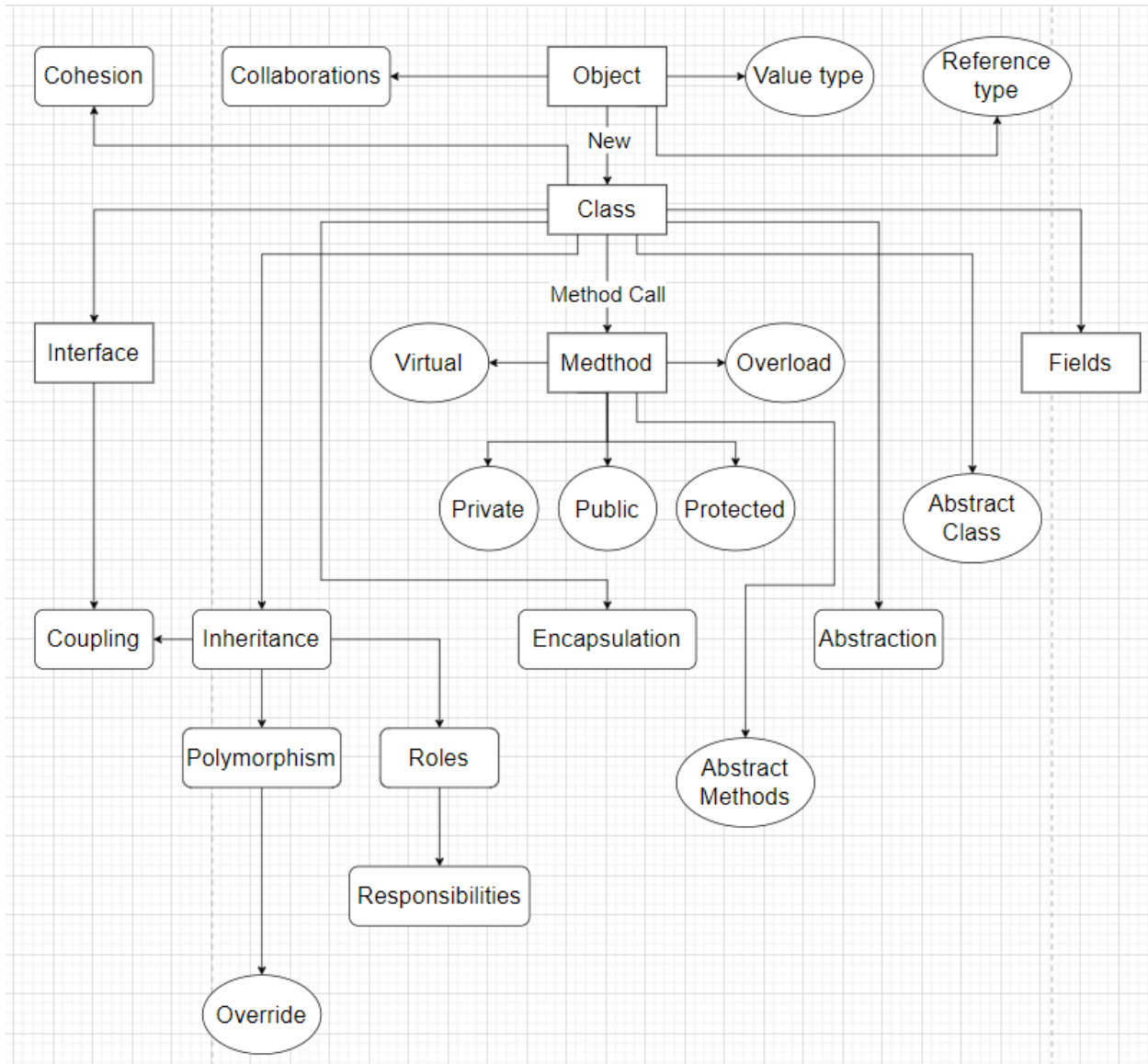


Key Object Oriented Concepts Report

Nguyen Duc Thang

104776473

- I. Below is the concept map that demonstrate how four key principles in OOP (Encapsulation, Polymorphism, Abstraction, Inheritance) interact with others concepts and artefacts in programming.



- II. There are 4 fundamental principles: Abstraction, Encapsulation, Inheritance and Polymorphism
 - 1. Abstraction: This concept refers to providing enough necessary information for the user to use the application that the class have. For instance a class can have functions and the user only need to know the relevant data for that function to work.
 - 2. Encapsulation: This is a technique of packaging data so that the unimportant data is hidden. For example when building a class you have to implement scope for variables so that the user can't access important data.
 - 3. Inheritance: This means that a class can contains methods and properties from another class. For example you can create sub classes that have all the information from the base class but you can extend the sub classes functionality by adding more methods or property.
 - 4. Polymorphism: This concept give you the flexibility of treating an object, or treating a child class as the parent class. An example of this is when you want to create a drawing method that can draw different shapes in different classes.

- III. Three examples where Exception Handling should be used
 - 1. When dealing with file when the file is corrupted, missing, wrong location....
 - 2. When doing with calculation where some operation can lead to data type overflow.
 - 3. When working with user input when the input format conflict with the operation format such as email without @ or incomplete phone number, wrong datatype...