A* ALGORITHM JAVA PROJECT

DHRUV GUPTA

2020A7PS1680P

- **Encapsulate what Varies**

Encapsulation is followed partially. In Main class, integer dist[] ,Set<Integer> settled and PriorityQueue<Node> pq follow encapsulation because they are private and thus they can only be accessed using class functions. In Node class int node and int cost are public so they don't follow encapsulation.

- **Favour composition over Inheritance**

This is not followed in the code but by achieving polymorphic behaviour and code reuse by their composition rather than inheritance from a base or parent class. As inheritance should be used when subclasses are there so in my code in future further application of principle of Favour composition over inheritance is beneficial.

- **Classes should be open for extension and closed for modification**

Classes are not very freely open for extension but they are closed for modification. It is possible to freely add more buttons and features but it will require some considerable amount of change in the code.

- **Depend on abstraction, do not depend on concrete classes**

No, since we are creating objects of the classes in the program , we require concrete classes. There is no abstraction as all the classes

used in the program have their objects created somewhere or another.

DESIGN PATTERNS

- PROTOTYPE PATTERN

The prototype pattern is used when the Object creation is a costly affair and requires a lot of time and resources and you have a similar object already existing. So this pattern provides a mechanism to copy the original object to a new object and then modify it according to our needs. This pattern uses java cloning to copy the object.

Also the prototype design pattern mandates that the Object which you are copying should provide the copying feature. It should not be done by any other class.

So as in my code I have to create object two times when entering the distance for two cities , therefore it will be beneficial touse prototype pattern in my code.