Krutarth Patel
B00835794
CSCI5308
Individual Assignment

Patterns Used:

CREATIONAL

1. Abstract Factory

Purpose: To create objects for the states

Benefit:

1. Creation of state objects without depending on their concrete version
2. Ensures code flexibility as it hides the concrete implementation and we can swap the states anytime with some another implementation without affecting the code

2. Singleton

Purpose: To create single instance of AbstractFactory, TrackRiskModifiers, and TrackSymptoms class

Benefit:

1. Only one instance is shared among all the classes which is exactly what the classes mentioned above requires

BEHAVIOURAL

1. State

Purpose: To get rid of switch-case statements and code duplication by delegating the behaviour of state-specific code in their corresponding states

Benefit:

1. Offers an ability to change behaviour of the program at runtime based on the state that is being executed
2. Ensures code maintainability because new states can be added without affecting the currently existing ones when new RiskModifiers are introduced
3. Ensures code flexibility and adaptability. If risk level condition(s) for a risk modifier is/are changed, we can easily accommodate it in its corresponding state class without affecting the other states