Financial Forecasting

Code:

```
interface ForecastStrategy {
    void forecast();
}

class ShortTermForecast implements ForecastStrategy {
    public void forecast() {
        System.out.println("Performing short-term financial forecasting (1-3 months).");
    }
}

class MidTermForecast implements ForecastStrategy {
    public void forecast() {
        System.out.println("Performing mid-term financial forecasting (6-12 months).");
    }
}

class LongTermForecast implements ForecastStrategy {
    public void forecast() {
        System.out.println("Performing long-term financial forecasting (1-5 years).");
    }
}

class ForecastFactory {
    public ForecastStrategy getForecastStrategy(String type) {
        if (type == null) return null;

        if (type.equalsIgnoreCase("SHORT")) {
            return new ShortTermForecast();
        } else if (type.equalsIgnoreCase("MID")) {
```

```
            return new MidTermForecast();
        } else if (type.equalsIgnoreCase("LONG")) {
            return new LongTermForecast();
        }
        return null;
    }
}

public class Main {
    public static void main(String[] args) {
        ForecastFactory factory = new ForecastFactory();

        ForecastStrategy forecast1 = factory.getForecastStrategy("SHORT");
        forecast1.forecast();

        ForecastStrategy forecast2 = factory.getForecastStrategy("MID");
        forecast2.forecast();

        ForecastStrategy forecast3 = factory.getForecastStrategy("LONG");
        forecast3.forecast();
    }
}
```

Output:

```
Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

D:\cognizant\week 1\Data Structure and algorithm\Financial forecasting>javac *.java

D:\cognizant\week 1\Data Structure and algorithm\Financial forecasting>java Main
Performing short-term financial forecasting (1-3 months).
Performing mid-term financial forecasting (6-12 months).
Performing long-term financial forecasting (1-5 years).

D:\cognizant\week 1\Data Structure and algorithm\Financial forecasting>
```