

## E-commerce Platform Search Function:

### Code:

```
interface SearchStrategy {
    void search(String query);
}

class NameSearch implements SearchStrategy {
    public void search(String query) {
        System.out.println("Searching for product by name: " + query);
    }
}

class CategorySearch implements SearchStrategy {
    public void search(String query) {
        System.out.println("Searching for product by category: " + query);
    }
}

class PriceRangeSearch implements SearchStrategy {
    public void search(String query) {
        System.out.println("Searching for product in price range: " + query);
    }
}

class SearchFactory {
    public SearchStrategy getSearchStrategy(String type) {
        if (type == null) return null;

        if (type.equalsIgnoreCase("NAME")) {
            return new NameSearch();
        } else if (type.equalsIgnoreCase("CATEGORY")) {
            return new CategorySearch();
        } else if (type.equalsIgnoreCase("PRICE")) {
            return new PriceRangeSearch();
        }
        return null;
    }
}

public class Main {
```

```
public static void main(String[] args) {  
    SearchFactory factory = new SearchFactory();  
  
    SearchStrategy search1 = factory.getSearchStrategy("NAME");  
    search1.search("Laptop");  
  
    SearchStrategy search2 = factory.getSearchStrategy("CATEGORY");  
    search2.search("Electronics");  
  
    SearchStrategy search3 = factory.getSearchStrategy("PRICE");  
    search3.search("500-1000");  
}  
}
```

### Output:

```
Microsoft Windows [Version 10.0.26100.4351]  
(c) Microsoft Corporation. All rights reserved.  
  
D:\cognizant\week 1\Data Structure and algorithm\E-commerce>javac *  
  
D:\cognizant\week 1\Data Structure and algorithm\E-commerce>java Ma  
Searching for product by name: Laptop  
Searching for product by category: Electronics  
Searching for product in price range: 500-1000  
  
D:\cognizant\week 1\Data Structure and algorithm\E-commerce>
```