1.Implementing factory method

```
interface Platform {
    void showPlatform();
}
class AndroidPlatform implements Platform {
    public void showPlatform() {
        System.out.println("Platform: Android");
    }
}
class IOSPlatform implements Platform {
    public void showPlatform() {
        System.out.println("Platform: iOS");
    }
}
class WindowsPlatform implements Platform {
    public void showPlatform() {
        System.out.println("Platform: Windows");
    }
}
class PlatformFactory {
    public static Platform getPlatform(String type) {
        if (type.equalsIgnoreCase("android")) {
            return new AndroidPlatform();
        } else if (type.equalsIgnoreCase("ios")) {
            return new IOSPlatform();
        } else if (type.equalsIgnoreCase("windows")) {
            return new WindowsPlatform();
        } else {
            return null;
        }
    }
}

public class FactMethod {
    public static void main(String[] args) {
        Platform p1 = PlatformFactory.getPlatform("android");
        if (p1 != null) p1.showPlatform();

        Platform p2 = PlatformFactory.getPlatform("ios");
        if (p2 != null) p2.showPlatform();

        Platform p3 = PlatformFactory.getPlatform("windows");
        if (p3 != null) p3.showPlatform();

        Platform p4 = PlatformFactory.getPlatform("linux");
        if (p4 != null) {
            p4.showPlatform();
        } else {
            System.out.println("Platform: Unknown");
        }
    }
}
```

OUTPUT:

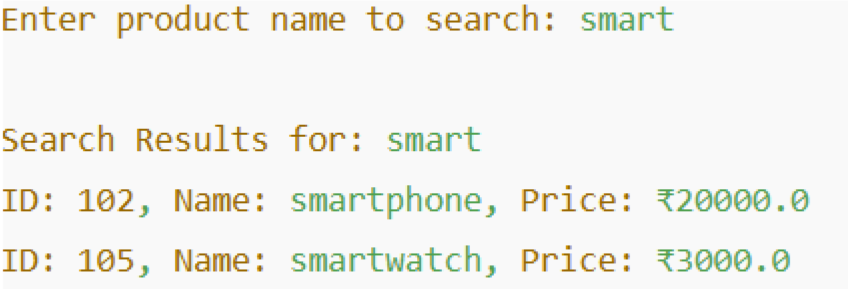
```
Platform: Android
Platform: iOS
Platform: Windows
```

Platform: Unknown

3.E-commerce platform Search Function

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
class Product {
    private int id;
    private String name;
    private double price;

    public Product(int id, String name, double price) {
        this.id = id;
        this.name = name.toLowerCase();
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", Price: ₹" + price);
    }
}
class ProductSearch {
    private List<Product> productList;

    public ProductSearch() {
        productList = new ArrayList<>();
        productList.add(new Product(101, "Laptop", 50000));
        productList.add(new Product(102, "Smartphone", 20000));
        productList.add(new Product(103, "Headphones", 1500));
        productList.add(new Product(104, "Shoes", 2500));
        productList.add(new Product(105, "Smartwatch", 3000));
        productList.add(new Product(106, "Backpack", 800));
    }

    public void search(String keyword) {
        keyword = keyword.toLowerCase();
        boolean found = false;

        System.out.println("\nSearch Results for: " + keyword);
        for (Product p : productList) {
            if (p.getName().contains(keyword)) {
                p.display();
                found = true;
            }
        }

        if (!found) {
            System.out.println("No products found.");
        }
    }
}
public class ECommerceSearch {
    public static void main(String[] args) {
        Scanner = new Scanner(System.in);

        ProductSearch searchEngine = new ProductSearch();


        System.out.print("Enter product name to search: ");

        String input = scanner.nextLine();


        searchEngine.search(input);

        scanner.close();

    }

}
```

Output:

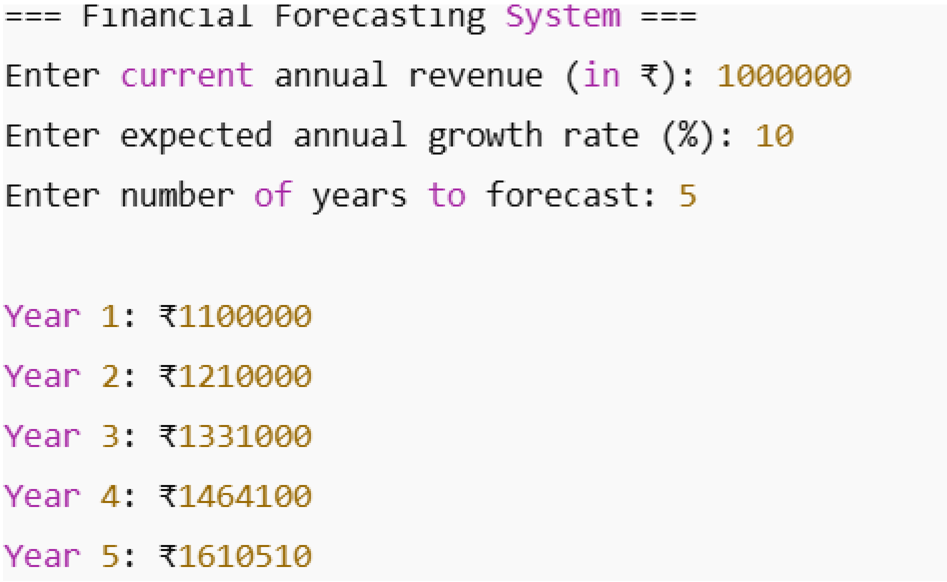


```
Enter product name to search: smart

Search Results for: smart
ID: 102, Name: smartphone, Price: ₹20000.0
ID: 105, Name: smartwatch, Price: ₹3000.0
```

4.Financial Forecasting

```
import java.util.Scanner;

import java.text.DecimalFormat;

public class FinancialForecasting {

  public static void main(String[] args) {

        Scanner = new Scanner(System.in);

        DecimalFormat df = new DecimalFormat("#.##");

        System.out.println("=== Financial Forecasting System ===");

        System.out.print("Enter current annual revenue (in ₹): ");

        double currentRevenue = scanner.nextDouble();
        System.out.print("Enter expected annual growth rate (%): ");
        double growthRate = scanner.nextDouble();
        System.out.print("Enter number of years to forecast: ");
        int years = scanner.nextInt();
        System.out.println("\n--- Revenue Forecast ---");
        for (int i = 1; i <= years; i++) {
            currentRevenue += currentRevenue * (growthRate / 100);
            System.out.println("Year " + i + ": ₹" + df.format(currentRevenue));
        }
        scanner.close();
    }
}
```

OUTPUT:



```
=== Financial Forecasting System ===
Enter current annual revenue (in ₹): 1000000
Enter expected annual growth rate (%): 10
Enter number of years to forecast: 5

Year 1: ₹1100000
Year 2: ₹1210000
Year 3: ₹1331000
Year 4: ₹1464100
Year 5: ₹1610510
```