Name: SASHMITHA R

Reg No: 727722EUCD042

Exercise 2: E-commerce Platform Search Function

Product.java

```
package Q2;

public class Product implements Comparable<Product>{

    int productId;

    String productName;

    String category;

    public Product(int productId, String productName, String category)

    {

        this.productId = productId;

        this.productName = productName;

        this.category = category;

    }

    @Override

    public int compareTo(Product p)

    {

        return this.productName.compareToIgnoreCase(p.productName);

    }

    @Override

    public String toString()

    {

        return productId + " " + productName + " " + category;

    }

}
```

SearchProduct.java

```
package Q2;

public class SearchProduct {

	public static Product linearSearch(Product[] pro, String str)  {

		for(Product p : pro)  {

			if(p.productName.equalsIgnoreCase(str))  {

				return p;

			}

		}

		return null;

	}

	public static Product binarySearch(Product[] pro, String str)  {

		int l = 0;

		int r = pro.length-1;

		while(l <= r)

		{

			int mid = l + (r-l)/2;

			int res = pro[mid].productName.compareToIgnoreCase(str);

			if(res == 0)  {

				return pro[mid];

			}

			else if(res < 0)  {

				l = mid+1;

			}

			else {

				r = mid-1;

			}

		}

		return null;

	}

}
```

Main.java

```
package Q2;

import java.util.*;

public class Main {

        public static void main(String[] args)

        {

                Product[] pro = {

                         new Product(101, "Laptop", "Electronics"),

                        new Product(102, "Chair", "Furniture"),

                        new Product(103, "Phone", "Electronics"),

                        new Product(104, "Shoes", "Footwear")

                };

    Product resultLinear = SearchProduct.linearSearch(pro, "Phone");

    System.out.println("Linear Search: " + (resultLinear != null ? resultLinear : "Not Found"));


    Arrays.sort(pro);


    Product resultBinary = SearchProduct.binarySearch(pro, "Phone");

    System.out.println("Binary Search: " + (resultBinary != null ? resultBinary : "Not Found"));

  }

}
```

Output:

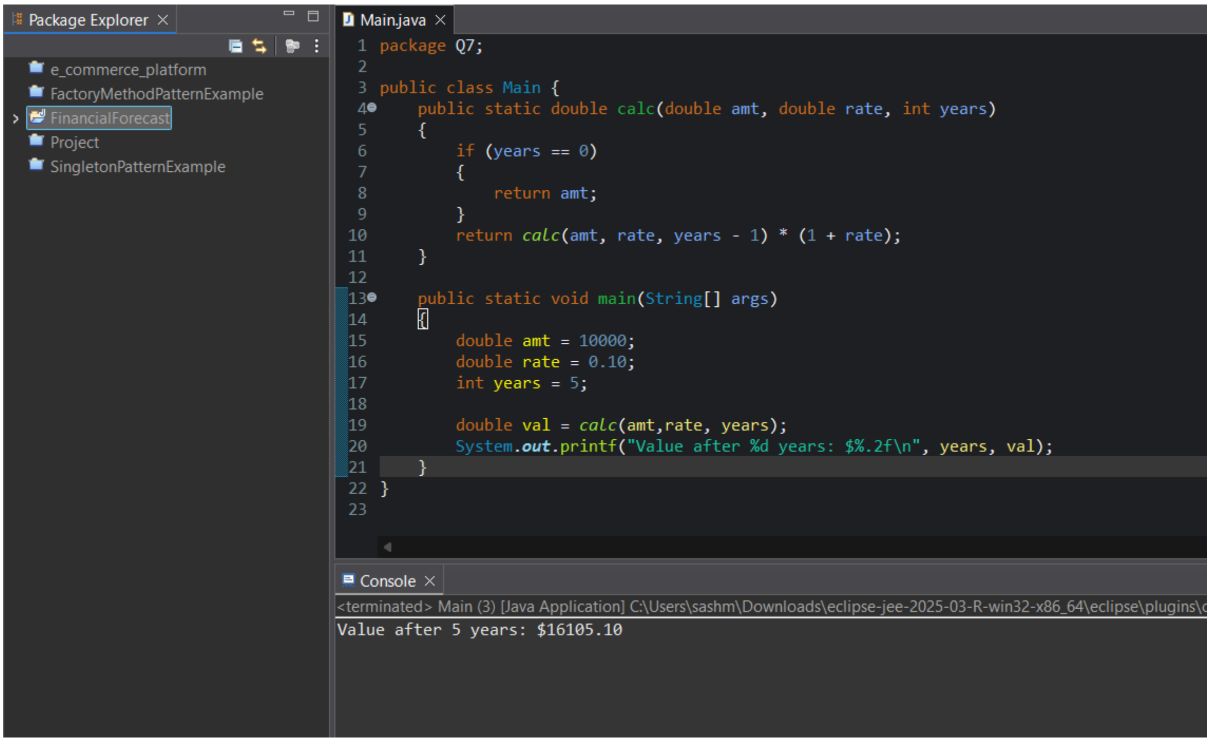
Linear Search: 103 Phone Electronics

Binary Search: 103 Phone Electronics

Analysis:

- If the dataset is small or unsorted, linear search is acceptable (O(n)).
- Binary Search is more suitable if products can be sorted and search is frequent, since it offers better performance (O(log n)).

Exercise 7: Financial Forecasting



Analysis:

Time Complexity

- Recursive Depth = n (number of years)
- Work per Call = Constant (O(1))
- Total Time Complexity = O(n)

Optimization Techniques

1. Convert to Iteration

- Best and simplest optimization.
- Avoids recursion overhead and stack usage.
- Time Complexity: O(n)
- Space Complexity: O(1)

2. Memoization

- Time Complexity: O(n)
- Space Complexity: O(n)

3. Mathematical Formula

- Best for compound growth
- Time Complexity: O(1)
- Space Complexity: O(1)