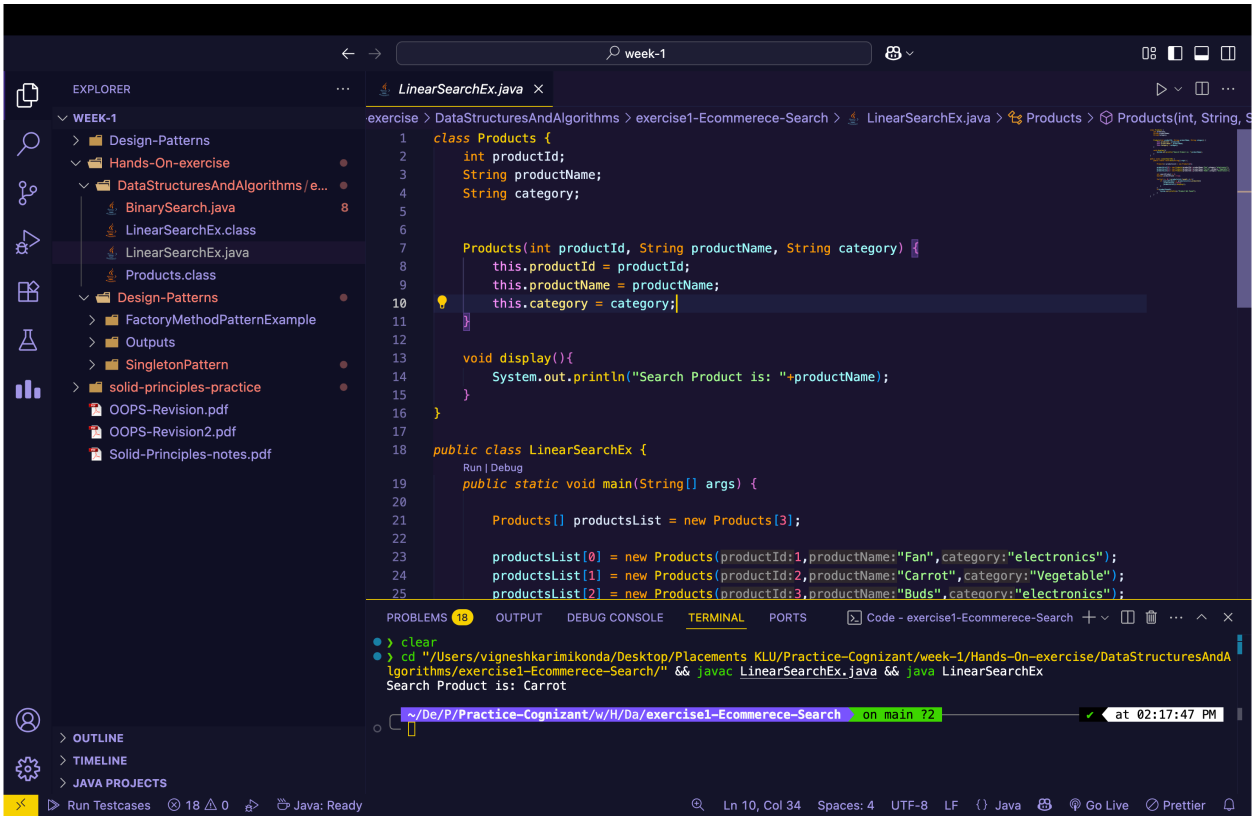
Analysis of Exercise 1 Linear & Binary Search:

Linear Search:

In Linear search the for Loop will run until to last index of array.

So if array size is 6 then the for loop will start from 0 to 5 and check if product is there or not

so **Time Complexity is o(n)**



```java
class Products {
    int productId;
    String productName;
    String category;


    Products(int productId, String productName, String category) {
        this.productId = productId;
        this.productName = productName;
        this.category = category;
    }

    void display(){
        System.out.println("Search Product is: "+productName);
    }
}

public class LinearSearchEx {
    Run | Debug
    public static void main(String[] args) {

        Products[] productsList = new Products[3];

        productsList[0] = new Products(productId:1,productName:"Fan",category:"electronics");
        productsList[1] = new Products(productId:2,productName:"Carrot",category:"Vegetable");
        productsList[2] = new Products(productId:3,productName:"Buds",category:"electronics");
```

```
❯ clear
❯ cd "/Users/vigneshkarimikonda/Desktop/Placements KLU/Practice-Cognizant/week-1/Hands-On-exercise/DataStructuresAndA
lgorithms/exercise1-Ecommerece-Search/" && javac LinearSearchEx.java && java LinearSearchEx
Search Product is: Carrot

~/De/P/Practice-Cognizant/w/H/Da/exercise1-Ecommerece-Search  on main ?2 ──────────── ✔  at 02:17:47 PM
```

Binary Search :

In Binary Search the Time complexity is **o(log n)**

Here,

at starting we take low = 0 , high = n-1

we will repeat while until high>=low

At staring we will calculate mid

next there are 3 cases:

1- if a[mid] == target then return true

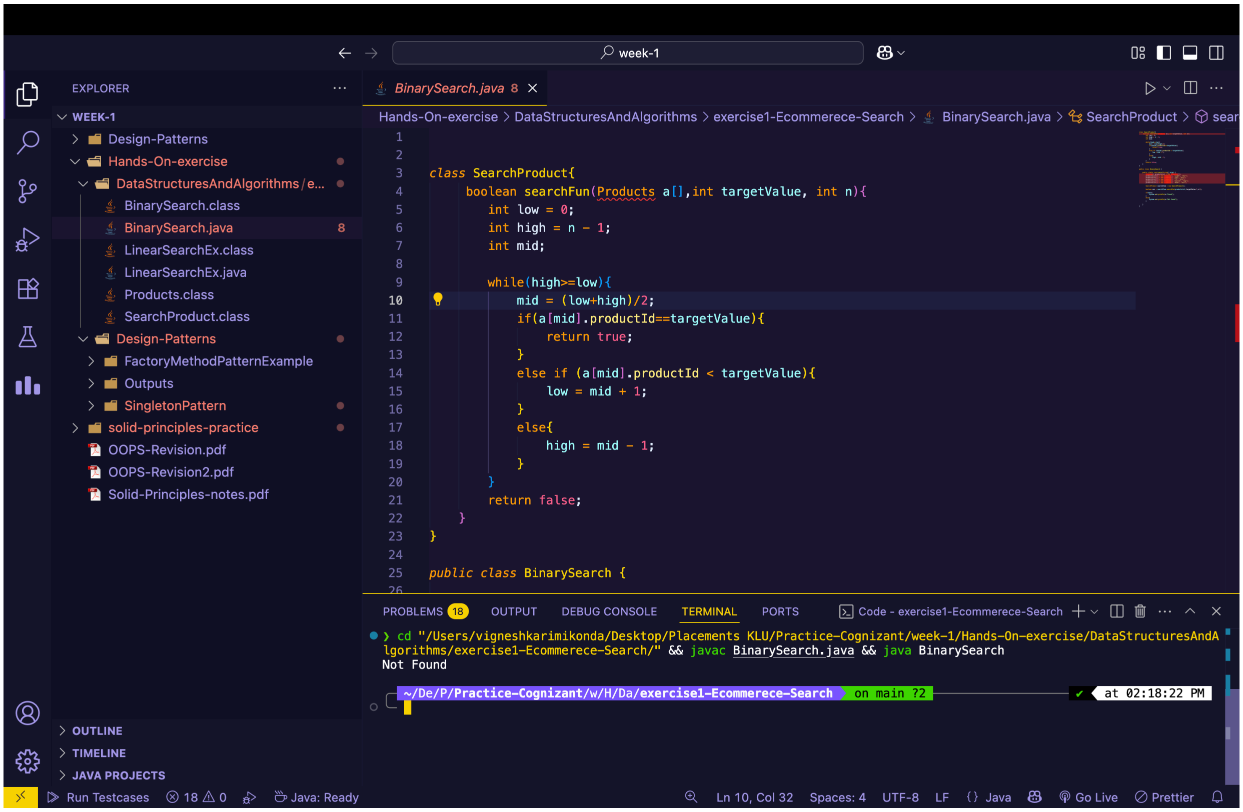
2- if a[mid]<target then , low = mid + 1;

3- if a[mid]>target then , high = mid – 1

we will repeat until then while Break

IMP : in Each iteration we eliminate search space by half

so the Time Complexity is: **o(Log n)**



```java


class SearchProduct{
     boolean searchFun(Products a[],int targetValue, int n){
        int low = 0;
        int high = n - 1;
        int mid;

        while(high>=low){
            mid = (low+high)/2;
            if(a[mid].productId==targetValue){
                return true;
            }
            else if (a[mid].productId < targetValue){
                low = mid + 1;
            }
            else{
                high = mid - 1;
            }
        }
        return false;
    }
}

public class BinarySearch {
```

```
❯ cd "/Users/vigneshkarimikonda/Desktop/Placements KLU/Practice-Cognizant/week-1/Hands-On-exercise/DataStructuresAndA
lgorithms/exercise1-Ecommerece-Search/" && javac BinarySearch.java && java BinarySearch
Not Found

 ~/De/P/Practice-Cognizant/w/H/Da/exercise1-Ecommerece-Search  on main ?2 ─────────── ✔  at 02:18:22 PM
```