

DATA STRUCTURES AND ALGORITHMS

MANDATORY HANDS-ON

Exercise 2: E-commerce Platform Search Function

Main.java:

```
import java.util.Scanner;

class 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 String toString(){

        return "["+productId+', '+productName+', '+category+']';

    }

}

class LinearSearch{

    public static Product search(Product[] products,int targetId){

        for(Product p:products){

            if(p.productId==targetId){

                return p;

            }

        }

        return null;

    }

}
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {  
        //System.out.println("Try programiz.pro");  
        Product[] products = {  
            new Product(101, "T-Shirt", "Fashion"),  
            new Product(102, "Shoes", "Footwear"),  
            new Product(103, "Watch", "Accessories"),  
            new Product(104, "Laptop", "Electronics"),  
            new Product(105, "Kurta Set", "Clothing"),  
            new Product(106, "Bangles", "Jewellery"),  
        };  
        Scanner s = new Scanner(System.in);  
        int searchId;  
        System.out.println("Enter the search ID of the product : ");  
        searchId=s.nextInt();  
        Product result = LinearSearch.search(products,searchId);  
        if(result!=null){  
            System.out.println("Product Found : "+result);  
        }  
        else{  
            System.out.println("The Product with id "+searchId+" not found");  
        }  
    }  
}
```

OUTPUT:

The screenshot shows the Programiz Online Java Compiler interface. The code editor on the left contains the following Java code:

```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3 import java.util.Scanner;
4 class Product{
5     int productId;
6     String productName;
7     String category;
8     public Product(int productId,String productName, String category
9     ){
10         this.productId=productId;
11         this.productName=productName;
12         this.category=category;
13     }
14     @Override
15     public String toString(){
16         return "["+productId+","+productName+","+category+"]";
17     }
18 }
19 class LinearSearch{
20     public static Product search(Product[] products,int targetId){
21         for(Product p:products){
22             if(p.productId==targetId){
23                 return p;
24             }
25         }
26     }
27 }
```

The output panel on the right shows the execution results:

```
Enter the search ID of the product :
103
Product Found : [103,Watch,Accessories]

=== Code Execution Successful ===
```

The screenshot shows the Programiz Online Java Compiler interface. The code editor on the left contains the following Java code:

```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3 import java.util.Scanner;
4 class Product{
5     int productId;
6     String productName;
7     String category;
8     public Product(int productId,String productName, String category
9     ){
10         this.productId=productId;
11         this.productName=productName;
12         this.category=category;
13     }
14     @Override
15     public String toString(){
16         return "["+productId+","+productName+","+category+"]";
17     }
18 }
19 class LinearSearch{
20     public static Product search(Product[] products,int targetId){
21         for(Product p:products){
22             if(p.productId==targetId){
23                 return p;
24             }
25         }
26     }
27 }
```

The output panel on the right shows the execution results:

```
Enter the search ID of the product :
107
The Product with id 107 not found

=== Code Execution Successful ===
```

Exercise 7: Financial Forecasting

Forecasting.java

```
//to calculate future value using  $FV = PV \cdot (1+r)^n$ 

import java.util.*;

public class Forecasting
{
    public static double forecast(double presentValue, double rate, int years){
        if(years==0){
            return presentValue;
        }
        return forecast(presentValue,rate,years-1)*(1+rate);
    }

    public static void main(String[] args) {
        System.out.println("Enter the present Value : ");
        Scanner s = new Scanner(System.in);
        double presentValue=s.nextDouble();
        System.out.println("Enter the annual growth rate : ");
        double rate = s.nextDouble(); //annual growth rate
        System.out.print("Enter the no. of years : ");
        int years=s.nextInt();
        double futureValue=forecast(presentValue,rate,years);
        System.out.printf("Future value after %d years : $%.3f\n", years,
futureValue);
        //System.out.println("Future value after "+years+" years : $" +futureValue);
    }
}
```

OUTPUT:

The screenshot shows a web browser window with the URL `onlinegdb.com/online_java_compiler#`. The browser's address bar and tabs are visible. Below the browser window is a code editor with a dark background and light-colored text. The code is a Java program named `Forecasting.java`. It defines a recursive method `forecast` that calculates the future value of an investment based on the present value, annual growth rate, and number of years. The `main` method prompts the user to enter these three values and then prints the calculated future value. The output of the program is displayed in a console window at the bottom of the editor, showing the user's input and the resulting future value of \$31531.050 after 7 years at a 0.17 growth rate.

```
12 public static double forecast(double presentValue, double rate, int years){
13     if(years==0){
14         return presentValue;
15     }
16     return forecast(presentValue,rate,years-1)*(1+rate);
17 }
18 public static void main(String[] args) {
19     System.out.println("Enter the present Value : ");
20     Scanner s = new Scanner(System.in);
21     double presentValue=s.nextDouble();
22     System.out.println("Enter the annual growth rate : ");
23     double rate = s.nextDouble(); //annual growth rate
24     System.out.print("Enter the no. of years : ");
25     int years=s.nextInt();
26     double futureValue=forecast(presentValue,rate,years);
27     System.out.printf("Future value after %d years : %.3f\n",years,futureValue);
28     //System.out.println("Future value after "+years+" years : "+futureValue);
29 }
```

input

```
Enter the present Value :
10506
Enter the annual growth rate :
0.17
Enter the no. of years : 7
Future value after 7 years : $31531.050

...Program finished with exit code 0
Press ENTER to exit console.
```

OUTPUT

Enter the present Value :

10506

Enter the annual growth rate :

0.17

Enter the no. of years : 7

Future value after 7 years : \$31531.050