**Data structure and Algorithm Assignment:**

**1)Search for E-commerce**

ProductSearch.java :

class Product {

int id;

String name;

String cat;

Product(int prodId, String prodName, String category) {

id = prodId;

name = prodName;

cat = category;

}

}

public class ProductSearch {

static Product linear(Product[] arr, int searchId) {

for (int i = 0; i < arr.length; i++) {

if (arr[i].id == searchId) {

return arr[i];

}

}

return null;

}

static Product binary(Product[] arr, int searchId) {

int low = 0;

int high = arr.length - 1;

while (low <= high) {

int mid = (low + high) / 2;

if (arr[mid].id == searchId) {

return arr[mid];

} else if (arr[mid].id < searchId) {

low = mid + 1;

} else {

high = mid - 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] list = {

new Product(1, "Phone", "Tech"),

new Product(2, "Jacket", "Clothes"),

new Product(3, "Pen", "Stationery")

};

Product found = linear(list, 2);

System.out.println(found != null ? found.name : "Nada");

Product[] sorted = {

new Product(1, "Phone", "Tech"),

new Product(2, "Jacket", "Clothes"),

new Product(3, "Pen", "Stationery")

};

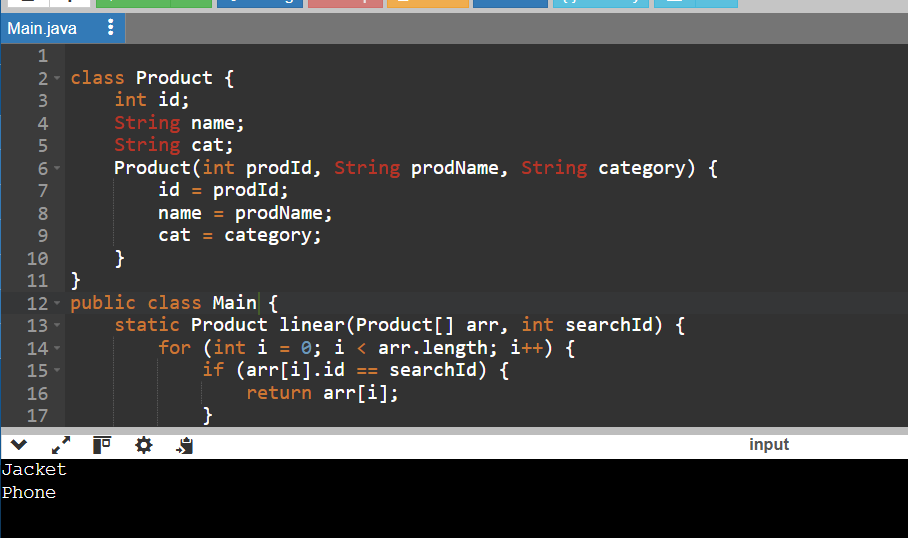
found = binary(sorted, 1);

System.out.println(found != null ? found.name : "Nada");

}

}

**Output:**

****

**2) Financial Forecasting:**

FinancialForecast.java

public class FinancialForecast {

static double calcFuture(double start, double rate, int yrs) {

if (yrs == 0) {

return start;

}

return calcFuture(start \* (1 + rate), rate, yrs - 1);

}

public static void main(String[] args) {

double amount = 1000;

double growth = 0.06;

int years = 2;

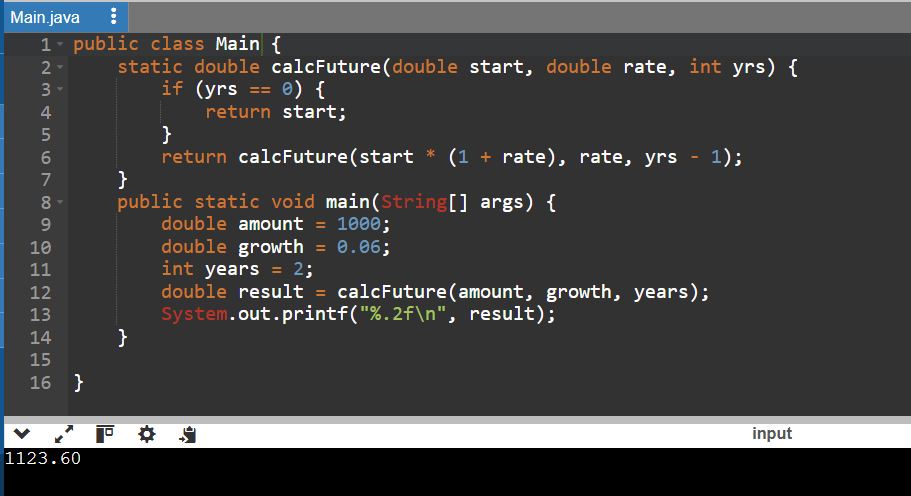
double result = calcFuture(amount, growth, years);

System.out.printf("%.2f\n", result);

}

}

**Output:**

****