**CTS DIGITAL NURTURE JAVA FSE**

**WEEK 1**

**DESIGN AND PATTERNS:**

***Singleton pattern:***

class Logger{

private static Logger i;

private Logger(){

System.out.println("Logger class");

}

public static Logger getInst(){

if(i==null){

i=new Logger();

}

return i;

}

public void Log(String m){

System.out.println("Log:"+m);

}}

public class Main

{

public static void main(String[] args) {

Logger l1=Logger.getInst();

l1.Log("Started");

Logger l2=Logger.getInst();

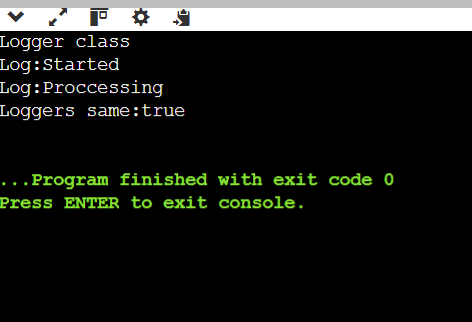
l2.Log("Proccessing");

System.out.println("Loggers same:"+(l1==l2));

}

}

**OUTPUT:**

****

***Factory Pattern:***

import java.util.Scanner;

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

System.out.println("Word document opened.");

}

}

class PdfDocument implements Document {

public void open() {

System.out.println("PDF document opened.");

}

}

class ExcelDocument implements Document {

public void open() {

System.out.println("Excel document opened.");

}

}

class DocumentFactory {

public static Document create(String type) {

switch (type.toLowerCase()) {

case "word":

return new WordDocument();

case "pdf":

return new PdfDocument();

case "excel":

return new ExcelDocument();

case "WORD":

return new WordDocument();

case "PDF":

return new PdfDocument();

case "EXCEL":

return new ExcelDocument();

default:

throw new IllegalArgumentException("Unknown type");

}

}

}

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter document type: ");

String input = sc.nextLine();

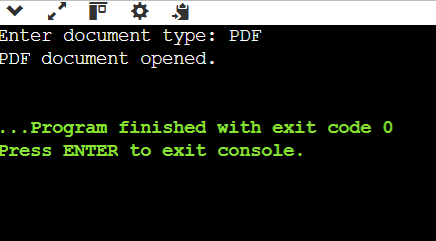
Document doc = DocumentFactory.create(input);

doc.open();

}

}

**OUTPUT:**

****

**DATA STRUCTURES AND ALGORITHM :**

***E-commerce platform search function:***

import java.util.\*;

class Product {

int productId;

String productName;

String category;

Product(int id, String name, String cat) {

productId = id;

productName = name;

category = cat;

}

public String toString() {

return productId + " - " + productName + " (" + category + ")";

}

}

class Main {

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop", "Electronics"),

new Product(2, "Shirt", "Clothing"),

new Product(3, "Book", "Education"),

new Product(4, "Phone", "Electronics")

};

Scanner sc = new Scanner(System.in);

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

String searchName = sc.nextLine();

boolean found = false;

for (Product p : products) {

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

System.out.println("Found: " + p);

found = true;

break;

}

}

if (!found)

System.out.println("Product not found.");

Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

System.out.println("\nUsing Binary Search:");

int low = 0, high = products.length - 1;

found = false;

while (low <= high) {

int mid = (low + high) / 2;

int compare = searchName.compareToIgnoreCase(products[mid].productName);

if (compare == 0) {

System.out.println("Found: " + products[mid]);

found = true;

break;

} else if (compare < 0) {

high = mid - 1;

} else {

low = mid + 1;

}

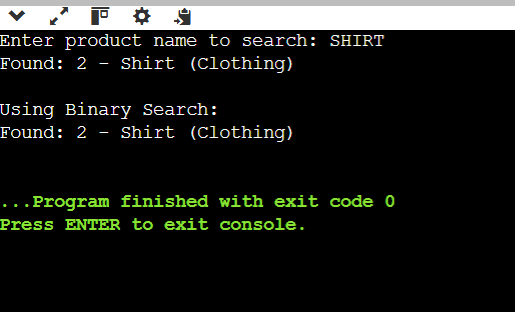
}

if (!found) System.out.println("Product not found.");

}

}

**OUTPUT:**

****

***Financial forecasting:***

import java.util.Scanner;

public class Main {

public static double fvRecursive(double p, double r, int y) {

if (y==0) {

return p;

}

return fvRecursive(p,r,y-1)\*(1+r);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the principal amount: ");

double p = scanner.nextDouble();

System.out.print("Enter the annual growth rate (e.g., 0.04 for 4%): ");

double r = scanner.nextDouble();

System.out.print("Enter the number of years: ");

int y = scanner.nextInt();

double fv = fvRecursive(p, r, y);

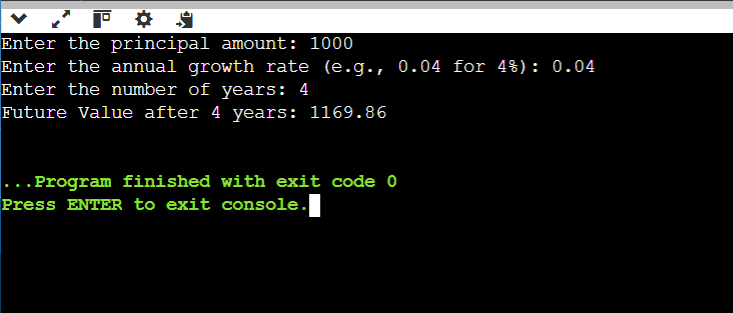
System.out.printf("Future Value after %d years: %.2f\n", y, fv);

scanner.close();

}

}

**OUTPUT:**

****