Slip 17

17.1

// Abstract Product: Shape interface

interface Shape {

void draw();

}

// Concrete Products: Circle, Square, Rectangle

class Circle implements Shape {

@Override

public void draw() {

System.out.println("Drawing Circle");

}

}

class Square implements Shape {

@Override

public void draw() {

System.out.println("Drawing Square");

}

}

class Rectangle implements Shape {

@Override

public void draw() {

System.out.println("Drawing Rectangle");

}

}

// Abstract Factory: ShapeFactory interface

interface ShapeFactory {

Shape createShape();

}

// Concrete Factories: CircleFactory, SquareFactory, RectangleFactory

class CircleFactory implements ShapeFactory {

@Override

public Shape createShape() {

return new Circle();

}

}

class SquareFactory implements ShapeFactory {

@Override

public Shape createShape() {

return new Square();

}

}

class RectangleFactory implements ShapeFactory {

@Override

public Shape createShape() {

return new Rectangle();

}

}

// Client Code

public class AbstractFactoryPatternExample {

public static void main(String[] args) {

// Create factories

ShapeFactory circleFactory = new CircleFactory();

ShapeFactory squareFactory = new SquareFactory();

ShapeFactory rectangleFactory = new RectangleFactory();

// Create shapes using the factories

Shape circle = circleFactory.createShape();

Shape square = squareFactory.createShape();

Shape rectangle = rectangleFactory.createShape();

// Draw shapes

circle.draw(); // Output: Drawing Circle

square.draw(); // Output: Drawing Square

rectangle.draw(); // Output: Drawing Rectangle

}

}

17.2

# Python program to implement Multiple Linear Regression

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset=pd.read\_csv('50\_Startups.csv')

x=dataset.iloc[:,:-1].values

y=dataset.iloc[:,-1].values

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

ct=ColumnTransformer(transformers=[('encoder',OneHotEncoder(),[3]) ],remainder='passthrough')

x=np.array(ct.fit\_transform(x)) print(x) from sklearn.model\_selection import train\_test\_split

17.3

const express = require('express');

const path = require('path');

const app = express();

const PORT = 3000;

// Serve static files from the 'public' directory

app.use(express.static(path.join(\_\_dirname, 'public')));

// Endpoint to trigger file download

app.get('/download', (req, res) => {

const filePath = path.join(\_\_dirname, 'public', 'example.txt');

// Set headers to make the browser prompt for download

res.setHeader('Content-Disposition', 'attachment; filename=example.txt');

res.sendFile(filePath);

});

// Start the server

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});