Slip 4

4.1

// Pizza interface

interface Pizza {

void prepare();

void bake();

void cut();

void box();

}

// Concrete Pizza class for NY style cheese pizza

class NyStyleCheesePizza implements Pizza {

@Override

public void prepare() {

System.out.println("Preparing NY Style Cheese Pizza");

}

@Override

public void bake() {

System.out.println("Baking NY Style Cheese Pizza");

}

@Override

public void cut() {

System.out.println("Cutting NY Style Cheese Pizza");

}

@Override

public void box() {

System.out.println("Boxing NY Style Cheese Pizza");

}

}

// Concrete Pizza class for Chicago style cheese pizza

class ChicagoStyleCheesePizza implements Pizza {

@Override

public void prepare() {

System.out.println("Preparing Chicago Style Cheese Pizza");

}

@Override

public void bake() {

System.out.println("Baking Chicago Style Cheese Pizza");

}

@Override

public void cut() {

System.out.println("Cutting Chicago Style Cheese Pizza");

}

@Override

public void box() {

System.out.println("Boxing Chicago Style Cheese Pizza");

}

}

// Pizza Store interface with the factory method createPizza()

interface PizzaStore {

Pizza createPizza(String type);

// Other methods like orderPizza() can be added here

}

// Concrete Pizza Store class for NY

class NyPizzaStore implements PizzaStore {

@Override

public Pizza createPizza(String type) {

if ("cheese".equalsIgnoreCase(type)) {

return new NyStyleCheesePizza();

}

// Add more pizza types as needed

return null;

}

}

// Concrete Pizza Store class for Chicago

class ChicagoPizzaStore implements PizzaStore {

@Override

public Pizza createPizza(String type) {

if ("cheese".equalsIgnoreCase(type)) {

return new ChicagoStyleCheesePizza();

}

// Add more pizza types as needed

return null;

}

}

public class PizzaStoreDemo {

public static void main(String[] args) {

PizzaStore nyStore = new NyPizzaStore();

PizzaStore chicagoStore = new ChicagoPizzaStore();

Pizza nyCheesePizza = nyStore.createPizza("cheese");

Pizza chicagoCheesePizza = chicagoStore.createPizza("cheese");

// Example of preparing and ordering pizzas

nyCheesePizza.prepare();

nyCheesePizza.bake();

nyCheesePizza.cut();

nyCheesePizza.box();

System.out.println();

chicagoCheesePizza.prepare();

chicagoCheesePizza.bake();

chicagoCheesePizza.cut();

chicagoCheesePizza.box();

}

}

4.2

# write python program to implement Simple Linear Regression for predicting house price

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import cross\_val\_predict

data = pd.read\_csv(r'kc\_house\_data.csv')

data.head(5);

print(data.shape)

f = ['price','bedrooms','bathrooms','sqft\_living','floors','condition','sqft\_abov e','sqft\_basement','yr\_built','yr\_renovated']

data = data[f]

print(data.shape)

data = data.dropna()

print(data.shape)

data.describe()

X=data[f[1:]]

y=data['price']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size =0.2,random\_state=42)

print(X\_train.shape)

print(X\_test.shape)

print(y\_train.shape

print(y\_test.shape)

lr=LinearRegression()

lr.fit(X\_train,y\_train)

print(lr.coef\_)

y\_test\_predict = lr.predict(X\_test)

print(y\_test\_predict.shape)

g = plt.plot((y\_test-y\_test\_predict),marker='o',linestyle="")

plt.show()

4.3

// Importing the 'readline' module to take user input

const readline = require('readline');

// Creating an interface to read input

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

// Prompt the user for input

rl.question('Enter a string: ', (inputString) => {

// Convert the input string to uppercase

const uppercaseString = inputString.toUpperCase();

// Print the result

console.log('Uppercase Output:', uppercaseString);

// Close the interface

rl.close();

});