Exercise 2 : Implementing the Factory Method Pattern

interface Assistant {

void assist();

}

class CodingAssistant implements Assistant {

public void assist() {

System.out.println("Let's debug that Java code together!");

}

}

class FitnessAssistant implements Assistant {

public void assist() {

System.out.println("Here's your 5-day gym plan for fat loss.");

}

}

class CareerCoachAssistant implements Assistant {

public void assist() {

System.out.println("Let's prep for your tech interviews!");

}

}

class AssistantFactory {

public Assistant getAssistant(String type) {

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

return new CodingAssistant();

} else if (type.equalsIgnoreCase("fitness")) {

return new FitnessAssistant();

} else if (type.equalsIgnoreCase("career")) {

return new CareerCoachAssistant();

}

return null;

}

}

public class Main {

public static void main(String[] args) {

AssistantFactory factory = new AssistantFactory();

Assistant a1 = factory.getAssistant("coding");

a1.assist();

Assistant a2 = factory.getAssistant("fitness");

a2.assist();

Assistant a3 = factory.getAssistant("career");

a3.assist();

}

}

Output:

