/\*

If class Algo having Process1, Process2, Process3 and Process4 methods. Currently class ConsumerA is using its instance and using all these 4 methods for data processing. There is another class ConsumerB where we need to provide access to only Process1 and Process3 and for class C we need to provide access to Process2 and Process4. What kind of design changes you need do to address this case without affecting current functionality that we have for class ConsumerA. (2)

\*/

using System;

public class Program

{

public static void Main()

{

Console.WriteLine("Hello World");

}

}

public class Algo

{

public int Process1(int a) { return a + 1; }

public int Process2(int a) { return a + 2; }

public int Process3(int a) { return a + 3; }

public int Process4(int a) { return a + 4; }

}

public class ConsumerA

{

private readonly Algo \_algo;

public ConsumerA(Algo obj)

{

\_algo = obj;

}

public int ProcessData(int data)

{

//Using all processing methods in Algo

var r = \_algo.Process1(data);

r = \_algo.Process2(r);

r = \_algo.Process3(r);

r = \_algo.Process4(r);

return r;

}

}

public class ConsumerB

{

private readonly Algo \_algo;

public ConsumerB(Algo obj)

{

\_algo = obj;

}

public int ProcessData(int data)

{

//Using it

var r = \_algo.Process1(data);

r = \_algo.Process2(r); // Should be restricted

r = \_algo.Process3(r);

r = \_algo.Process4(r); // Should be restricted

return r;

}

}

public class ConsumerC

{

private readonly Algo \_algo;

public ConsumerC(Algo obj)

{

\_algo = obj;

}

public int ProcessData(int data)

{

//Using it

var r = \_algo.Process1(data); // Should be restricted

r = \_algo.Process2(r);

r = \_algo.Process3(r);// Should be restricted

r = \_algo.Process4(r);

return r;

}

}