Give the outputs of the 2 programs, the only difference between them is the modifier on methodA in the classes Base6 and Base7.

a) Do not run the program yet. Put a number next to the methods that are called in the order they are called (several methods are never called in this example).

b) Write a explaination of how why you obtained your solution.

c) You may compile and run this code. If you answer does not match the results please explain.

public class Base6 {

private int x = 1;

// there is no dynamic dispatching

// for private instance methods, but

// normal rules apply to methods

// called on the implicit parameter

**private** void methodA(int u) {

methodB(x + u);

}

public void methodB(int a) {

methodC(x);

}

public void methodC(int v) {

System.out.println(3\*v);

}

public void test (int i) {

methodA(i);

}

}

public class Derived6 extends Base6 {

private int y = 3;

// cannot Override

public void methodA(int u) {

methodC(y + 2\*u);

}

@Override

public void methodB(int a) {

System.out.println(a);

}

@Override

public void methodC(int v) {

methodB(v);

}

}

public class Quiz6 {

public static void main(String[] args) {

Base6 b = new Derived6();

b.test(2);

}

}

public class Base7 {

private int x = 1;

**public** void methodA(int u) {

methodB(x + u);

}

public void methodB(int a) {

methodC(x);

}

public void methodC(int v) {

System.out.println(v);

}

public void test (int i) {

methodA(i);

}

}

public class Derived7 extends Base7 {

private int y = 3;

@Override

public void methodA(int u) {

methodC(y + 2\*u);

}

@Override

public void methodB(int a) {

System.out.println(a);

}

@Override

public void methodC(int v) {

methodB(v);

}

}

public class Quiz6a {

public static void main(String[] args) {

Base7 b = new Derived7();

b.test(2);

}

}