Question 2:

Write a Car class with a private field speed that stores the current speed as an integer. The constructor sets the initial speed (default = 0); if the provided value is negative, it must print “Invalid Speed.” and set the speed to 0. Implement an accelerate(amount) method that increases the speed only when amount > 0; otherwise, it should print “Invalid Amount.” Implement a brake(amount) method that decreases the speed only when amount > 0. If the requested deceleration would make the speed fall below 0, set the speed directly to 0. Finally, provide a getSpeed() method so external code can query the current speed while keeping the speed field properly encapsulated and protected.

Test Cases:

| **Tests** | **Results** |
| --- | --- |
| Car car1 = new Car(-10);  System.out.println(car1.getSpeed()); | Invalid Speed.  0 |
| Car car2 = new Car(0);  car2.accelerate(30);  System.out.println(car2.getSpeed()); | 30 |
| Car car3 = new Car(10);  car3.accelerate(-5);  System.out.println(car3.getSpeed()); | Invalid Amount.  10 |
| Car car4 = new Car(20);  car4.brake(25);  System.out.println(car4.getSpeed()); | 0 |
| Car car5 = new Car(20);  car5.brake(-5);  System.out.println(car5.getSpeed()); | Invalid Amount.  20 |
| int nearMax = Integer.MAX\_VALUE;  Car car6 = new Car(nearMax);  car6.accelerate(1);  System.out.println(car6.getSpeed()); | 2147483648 |
| Car car7 = new Car(0);  car7.brake(10);  System.out.println(car7.getSpeed()); | 0 |
| Car car8 = new Car(0);  car8.accelerate(50);  car8.brake(20);  System.out.println(car8.getSpeed()); | 30 |