

Assignment 5 Sample Test Case

Question 1

```
public static void testVehicle() {  
    //The Vehicle class is abstract. We cannot create objects from this class.  
    Therefore we cannot test anything  
}
```

Question 2

```
public static void testCar() {  
    // test case for constructors  
    try {  
        // exception  
        Car c1 = new Car(70,80);  
    } catch (BadSpeedSetting e) {  
        System.out.println(e.getMessage().equals("Speed cannot be greater than speed  
limit!"));  
    }  
    try {  
        // exception  
        Car c2 = new Car(70,-10);  
    } catch (BadSpeedSetting e) {  
        System.out.println(e.getMessage().equals("Speed cannot be negative!"));  
    }  
  
    // test case for accelerate, canFly and getSpeed methods  
    try {  
        Car c3 = new Car(100,80);  
        // test the canFly method  
        System.out.println(c3.canFly() == false);  
        c3.accelerate(10);  
        System.out.println(c3.getSpeed() == 90);  
        // test the accelerate with positive amount  
        c3.accelerate(-20);  
        System.out.println(c3.getSpeed() == 70);  
        // test the accelerate with negative amount  
        c3.accelerate(35); // exception  
    } catch (BadSpeedSetting e) {  
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");  
    } catch (ExceedSpeedLimit e) {  
        System.out.println(e.getMessage().equals("Current speed is 70. Accelerate 35  
will exceed the speed limit!"));  
    } catch (NotEnoughSpeed e) {  
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");  
    }  
  
    try {  
        Car c4 = new Car(100,80);  
        // exception  
        c4.accelerate(-90);  
    } catch (BadSpeedSetting e) {  
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");  
    } catch (ExceedSpeedLimit e) {  
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");  
    }
```

```

    } catch (NotEnoughSpeed e) {
        System.out.println(e.getMessage().equals("Current speed is "+ 80 + ", not
enough speed to decelerate!"));
    }
}

```

Question 3

```

public static void testAirCraft()
{
    // test case for constructors
    try {
        // Exception
        AirCraft a1 = new AirCraft(70,80,1000);
    } catch (BadSpeedSetting e) {
        System.out.println(e.getMessage().equals("Speed cannot be greater than speed
limit!"));
    } catch (BadAltitudeSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }
    try {
        // exception
        AirCraft a2 = new AirCraft(70,-10,1000);
    } catch (BadSpeedSetting e) {
        System.out.println(e.getMessage().equals("Speed cannot be negative!"));
    } catch (BadAltitudeSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }
    try {
        // exception
        AirCraft a3 = new AirCraft(700,200,-10);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (BadAltitudeSetting e)
    { System.out.println(e.getMessage().equals("Altitude cannot be negative!"));
    }

    // Test case for accelerate, canFly, getSpeed and getAltitude methods
    try {
        AirCraft a4 = new AirCraft(700,200,100);
        // test the getAltitude method
        System.out.println(a4.getAltitude()==100);
        // test the canFly method
        System.out.println(a4.canFly() == true);
        a4.accelerate(100);
        System.out.println(a4.getSpeed() == 300);
        // test the accelerate with positive amount
        a4.accelerate(-200);
        System.out.println(a4.getSpeed() == 100);
        // test the accelerate with negative amount
        // exception
        a4.accelerate(650);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (BadAltitudeSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (ExceedSpeedLimit e) {
        System.out.println(e.getMessage().equals("Current speed is 100. Accelerate
650 will exceed the speed limit!"));
    } catch (NotEnoughSpeed e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }
}

```

```

    }

    try {
        AirCraft a5 = new AirCraft(700,200,100);
        // exception
        a5.accelerate(-300);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (BadAltitudeSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (ExceedSpeedLimit e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (NotEnoughSpeed e) {
        System.out.println(e.getMessage().equals("Current speed is "+ 200 + ", not
enough speed to decelerate!"));
    }
}

```

Question 4

```

public static void testTrain() {
    // test case for constructors
    try {
        // exception
        Train t1 = new Train(350,400);
    } catch (BadSpeedSetting e) {
        System.out.println(e.getMessage().equals("Speed cannot be greater than speed
limit!"));
    }
    try {
        // exception
        Train t2 = new Train(350,-100);
    } catch (BadSpeedSetting e) {
        System.out.println(e.getMessage().equals("Speed cannot be negative!"));
    }

    // test case for accelerate, canFly and getSpeed methods
    try {
        Train t3 = new Train(350,300);
        // test the canFly method
        System.out.println(t3.canFly() == false);
        // test the getSpeed
        System.out.println(t3.getSpeed() == 300);
        // exception
        t3.accelerate(50);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (TrainSpeedChange e) {
        System.out.println(e.getMessage().equals("Do not try to accelerate or
decelerate the train!"));
    }
}

```

Question 5

```

public static void testManyVehicles()
{
    ManyVehicles mv = new ManyVehicles();
    // Add any kind of vehicles to the arraylist.
    try {
        Train t = new Train(350,300);
    }
}

```

```

        mv.addVehicle(t);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }

    try {
        Car c = new Car(100,80);
        mv.addVehicle(c);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }

    try {
        AirCraft a = new AirCraft(700,200,100);
        mv.addVehicle(a);
    } catch (BadSpeedSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    } catch (BadAltitudeSetting e) {
        System.out.println("BUG! THIS MUST NEVER HAPPEN!");
    }
    System.out.println(mv.calcAvgSpeed() == 193);
}

```