

Aşağıda verilen sınıf tanımlamalarına göre verilen ifadeleri “hatasız”, “derleme zamanı” veya “çalışma zamanı” hatası olacak şekilde 3 etiketle işaretleyiniz. Eğer ifade hatalı ise bu hatanın ne olduğunu ve nedenini yazınız.

```
public class Person {  
    public void birth() {}  
  
    public void takeHoliday() {}  
}  
  
public class GradStudent extends  
    Student {  
  
    public void writeThesis() {}  
  
}
```

```
public class Student extends Person {  
  
    public void takeExam() {}  
  
    public void takeHoliday() {}  
}  
  
public class Lecturer extends Person {  
    public void teach() {}  
}
```

- (a) Person person = **new** Student();
person.takeExam();
- (c) Student student = **new** Student();
Student g= (GradStudent)student;
- (d) Person person = **new** Student();

```
Lecturer lecturer = (Lecturer)person;
```

(e)

```
GradStudent student = new GradStudent();  
GradStudent grads = (Student)student;
```

(f)

```
Person person = new Lecturer();  
person.teach();
```

4. Aşağıda verilen sınıf tanımlarına göre aşağıdaki metod çağrılarının sonucunu yazınız.

```
public class Vehicle {
    public void m1() {
        System.out.println("Vehicle_1");
    }
}
class Car extends Vehicle {
    public void m1() {
        System.out.println("Car_1");
    }

    public void m3() {
        System.out.print("Car_3");
        super.m1();
    }
}
```

```
class SportsCar extends Car {
    public void m2() {
        System.out.println("SportsCar_2");
    }
}
class Maserati extends SportsCar {
    public void m2() {
        System.out.print("Maserati_2");
        super.m1();
    }
}
```

```
Object o = new Vehicle();
Vehicle v = new Car();
Car c = new SportsCar();
SportsCar s = new Maserati();
```

c.m1();	_____
s.m2();	_____
c.m3();	_____
s.m1();	_____
v.m2();	_____
v.m3();	_____
c.m2();	_____
s.m1();	_____
((Vehicle) o).m1();	_____
((Car) o).m1();	_____

3. Aşağıdaki programın vereceği çıktıyı yazınız

```
public class Suspicious {
    public static void main(String[] args) {
        try {
            try {
                System.out.print("A");
                bad();
            } catch (Exception e) {
                bad();
                System.out.print("B");
            } finally {
                System.out.print("C");
            }
        } catch (Exception e) {
            System.out.print("E");
        }
        System.out.print("G");
    }

    private static void bad() throws Exception {
        try {
            System.out.print("F");
            throw new Exception();
        } finally {
            System.out.print("D");
        }
    }
}
```