

public class Cat extends Pet

public Cat (String name)
super(name);

public String speak()
return "meow";

}

```
public class LoudDog extends Dog
```

```
{
```

```
    public LoudDog (String name)
```

```
    {
```

```
        super(name);
```

```
    }
```

```
    public String speak()
```

```
    {
```

```
        return super.speak() + " " + super.speak();
```

```
    }
```

```
}
```

```
public void allSpeak()
```

```
{
```

```
    for (int i=0; i < petList.size(); i++)
```

```
    {
```

```
        Pet hold = petList.get(i);
```

```
        System.out.println(hold.getName() + " " + hold.speak());
```

```
    }
```

```
}
```

**AP[®] Computer Science A
2004 SCORING GUIDELINES**

Question 2

Part A:	class Cat	2 pts
----------------	-----------	--------------

- +1/2 public class Cat extends Pet
- +1/2 Constructor correct (must call super)
- +1 speak method
 - +1/2 attempt (method header matches abstract method, OK if abstract left in)
 - +1/2 correct

Part B:	class LoudDog	3 pts
----------------	---------------	--------------

- +1/2 public class LoudDog extends Dog
- +1 Constructor correct (must call super)
- +1 1/2 speak method
 - +1 attempt (calls super.speak() *and* method header matches abstract method, OK if abstract left in)
 - +1/2 correct value returned

Part C:	Kennel - allSpeak	4 pts
----------------	-------------------	--------------

- +1 loop over petList
 - +1/2 attempt
 - +1/2 correct (must access petList)
- +1 1/2 get pet from petList (no deduction for missing downcast from petList)
 - +1/2 attempt
 - +1 correct (local variable must be type Pet)
- +1 1/2 print p.getName() and p.speak() for pet p (local variable not necessary)
 - +1/2 attempt (must have xxx.getName() or xxx.speak(), for some xxx)
 - +1 correct

Note: if done in-line with no local, no deduction for missing downcast.

-
- Usage:
- 1/2 public instance variable
 - 1 parent class name instead of super
 - 1/2 getName is overridden (other than super.getName) in part (a) and/or part (b)

(No deduction for other additional methods or constructors.)