Log on to **Socrative.com**
and join room **JMUAdkins**

Reece Adkins

CS 159 REVIEW, JAMES MADISON UNIVERSITY

ABSTRACT CLASSES AND INTERFACES

Consider this class...

```
public class LoudToy
{
    private int volume;

    public LoudToy(int volume)
    {
        this.volume = volume;
    }

    public void makeNoise()
    {
        System.out.println("Generic noise");
    }

    public int getVolume()
    {
        return volume;
    }
}
```

Consider this class…

```
public abstract class LoudToy
{
    private int volume;

    public LoudToy(int volume)
    {
        this.volume = volume;
    }

    public void makeNoise()
    {
        System.out.println("Generic noise");
    }

    public int getVolume()
    {
        return volume;
    }
}
```

How does adding this keyword change our class?

Log on to **Socrative.com** and join room **JMUAdkins**

WHAT IS AN ABSTRACT CLASS?

- A class that is not instantiated, but is extended by other classes.
  AccessSpecifier abstract class *ClassName*

WHAT IS AN ABSTRACT METHOD?

- A method that has no body and must be overridden in subclasses.
  AccessSpecifier abstract *ReturnType MethodName*(*ParameterList*);

Consider this class…

```
public abstract class LoudToy
{
    private int volume;

    public LoudToy(int volume)
    {
        this.volume = volume;
    }

    public void makeNoise()
    {
        System.out.println("Generic noise");
    }

    public int getVolume()
    {
        return volume;
    }
}
```

This is a generic method that needs to be overridden by subclasses.

Turn this method into an abstract method.

Log on to **Socrative.com** and join room **JMUAdkins**

Solution

```
public abstract class LoudToy
{
    private int volume;

    public LoudToy(int volume)
    {
        this.volume = volume;
    }

    public abstract void makeNoise();

    public int getVolume()
    {
        return volume;
    }
}
```

Log on to **Socrative.com** and join room **JMUAdkins**

Abstract Class Example (Continued)

```
public class ToyRobot extends LoudToy
{
    private int chargeLevel;

    public ToyRobot()
    {
        super(10);
        chargeLevel = 5;
    }

    @Override
    public void makeNoise()
    {
        System.out.println(“Beep Beep!”);
    }

    public int recharge()
    {
        chargeLevel = 10;
        System.out.println(“Charged Up”)!
    }
}
```

```
public class ToySheep extends LoudToy
{
    public ToySheep()
    {
        super(3);
    }


    @Override
    public void makeNoise()
    {
        System.out.println(“Baaa.”);
    }
}
```

WHAT IS AN INTERFACE?

- Specifies the behavior of a class.
- Looks similar to a class, except the keyword interface is used instead of the keyword class.
- public interface *InterfaceName*
  {
  (Method headers…)
  }

Consider this interface...

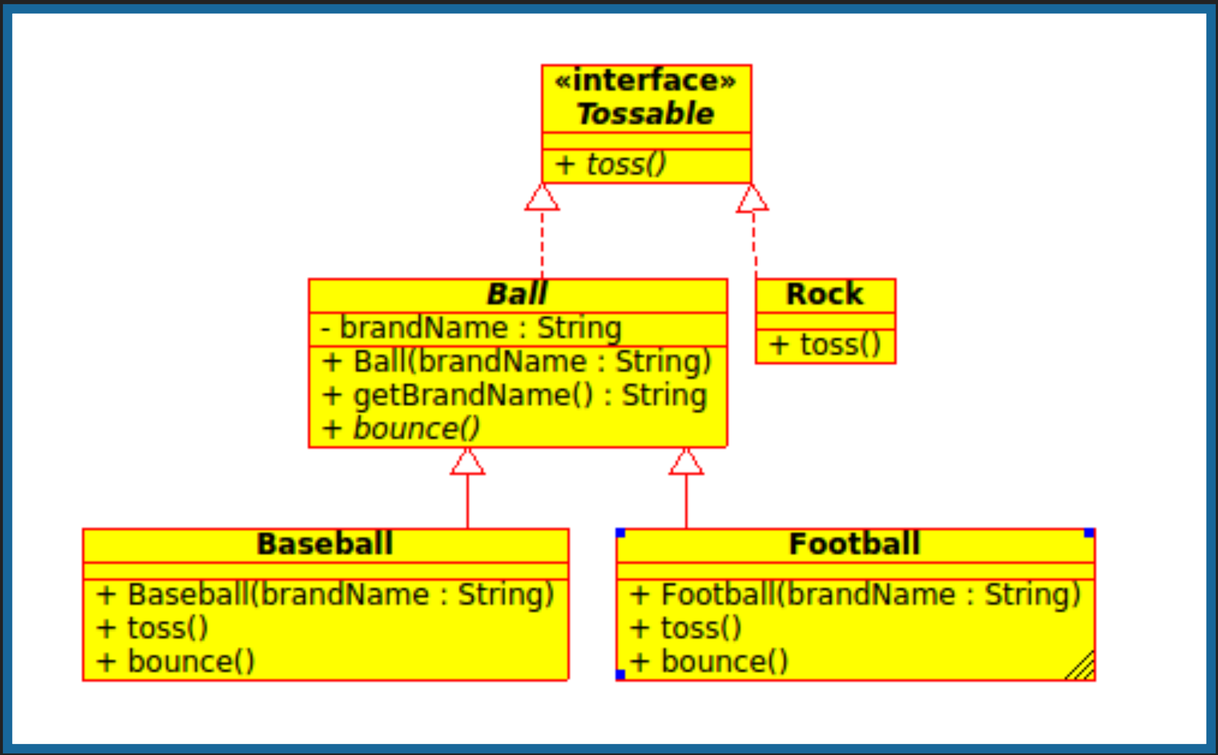
```
/**
 *  Any class that has an adjustable volume
 *  may implement this interface.
 */

public interface Audible
{
    int MAX_VOLUME = 10;

    int getVolume();

    void setVolume(int volume);
}
```

Consider the following UML Diagram...

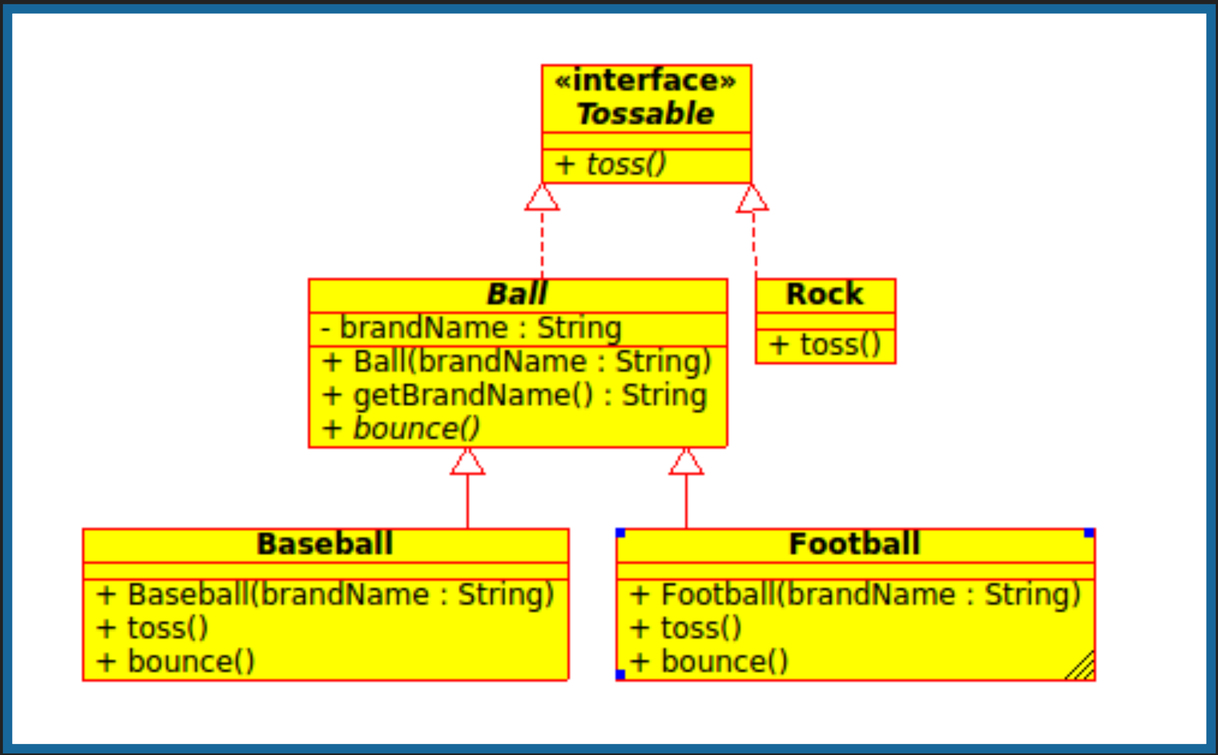


Indicate if the following line of code will compile or execute.

```
Ball ball = new Football("spalding");
```

Log on to **Socrative.com** and join room **JMUAdkins**

Consider the following UML Diagram...

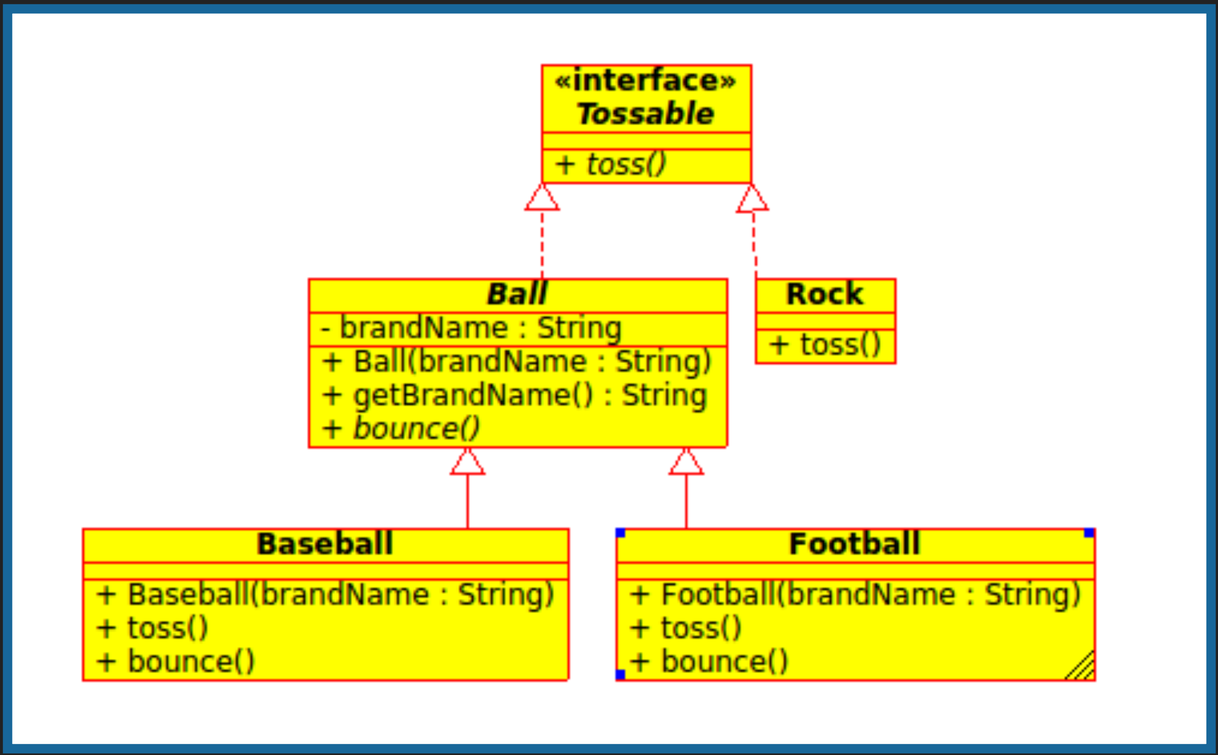


Indicate if the following line of code will compile or execute.

```
Ball ball = new Football("Spalding");
Baseball baseball = (Baseball)ball;
```

Log on to **Socrative.com** and join room **JMUAdkins**

Consider the following UML Diagram...

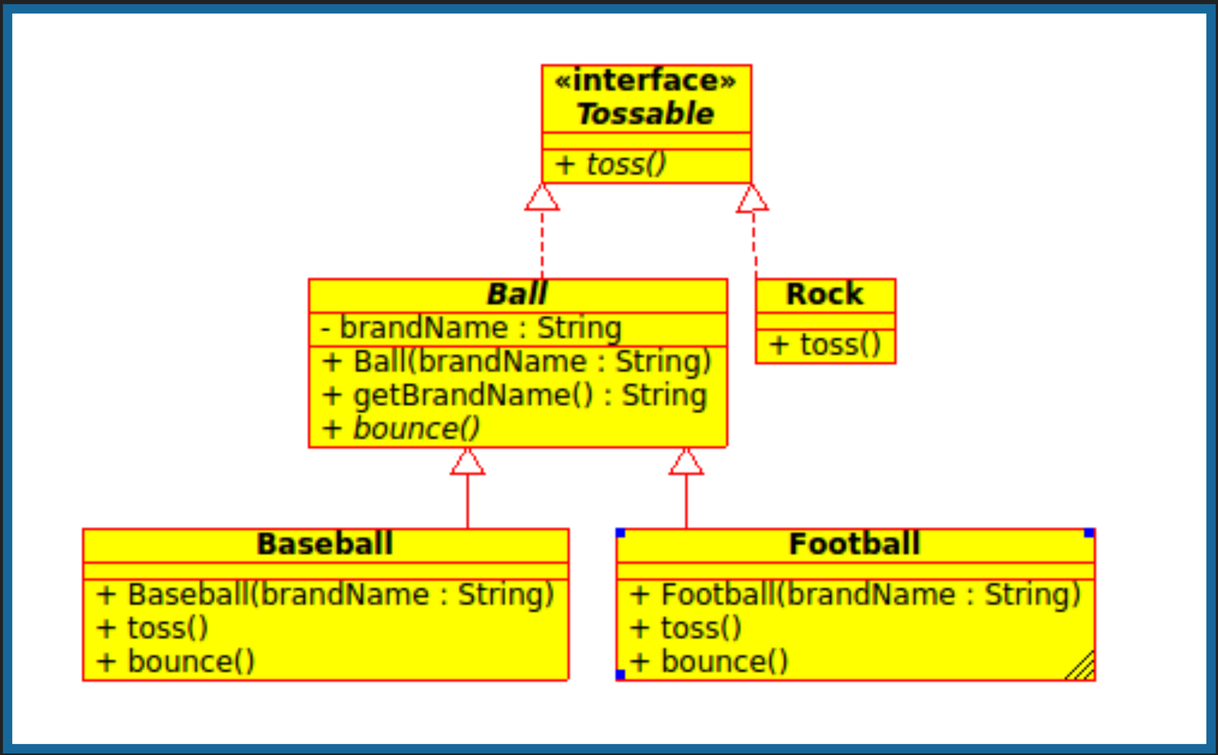


Indicate if the following line of code will compile or execute.

```
Object obj = new Baseball("spalding");
```

Log on to **Socrative.com** and join room **JMUAdkins**

Consider the following UML Diagram...

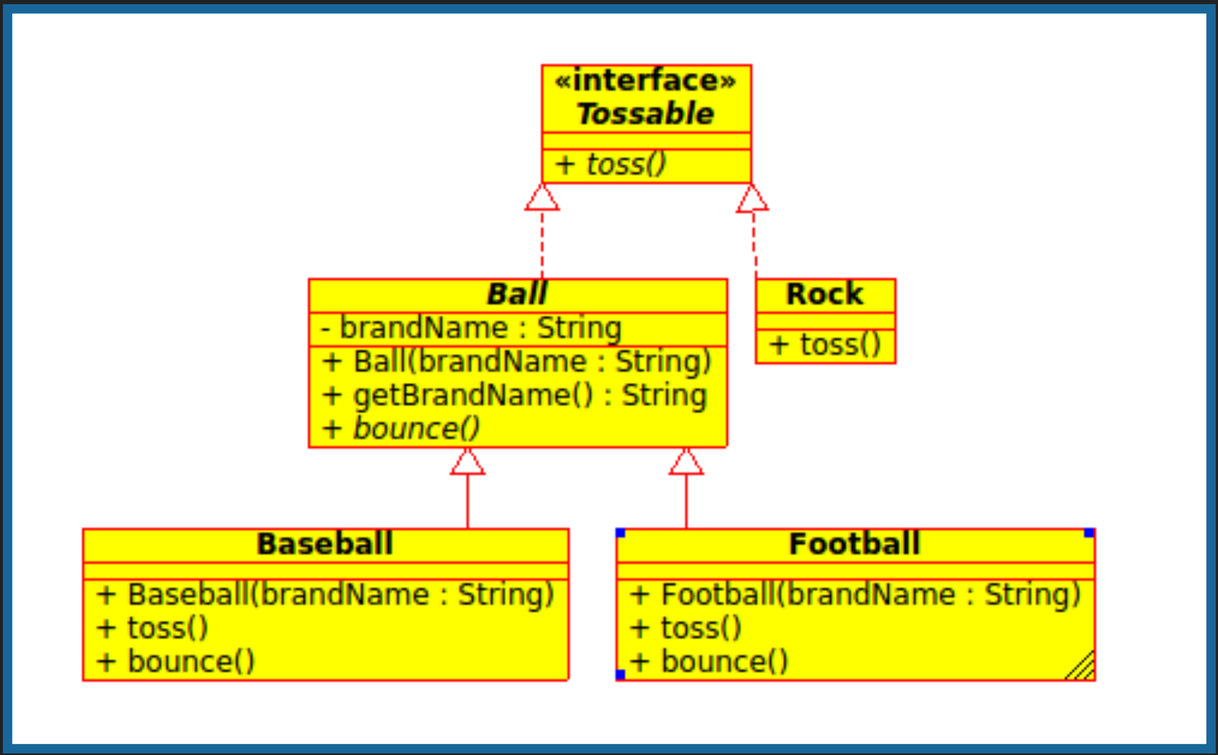


Indicate if the following line of code will compile or execute.

```
Object obj = new Baseball("spalding");
Tossable tossable = obj;
```

Log on to **Socrative.com** and join room **JMUAdkins**

Consider the following UML Diagram...



Indicate if the following line of code will compile or execute.

```
Tossable tossable = new Baseball("spalding");
Object obj = tossable;
```

Log on to **Socrative.com** and join room **JMUAdkins**