

★ Multiple Choice Practice: Inheritance & Polymorphism ★

15. Consider the following two classes.

```

public class Dog
{
    public void act()
    {
        System.out.print("run ");
        eat();
    }
    public void eat()
    {
        System.out.print("eat ");
    }
}
public class UnderDog extends Dog
{
    public void act()
    {
        super.act();
        System.out.print("sleep ");
    }
    public void eat()
    {
        super.eat();
        System.out.print("bark ");
    }
}
    
```

Assume that the following declaration appears in a class other than Dog.

*variable type*

```

Dog fido = new UnderDog();
    
```

*object type*

*run. eat bark sleep*

What is printed as a result of the call fido.act() ?

- (A) run eat
- (B) run eat sleep
- (C) run eat sleep bark
- (D) run eat bark sleep
- (E) Nothing is printed due to infinite recursion.

22. Consider the following declaration for a class that will be used to represent points in the  $xy$ -coordinate plane.

```
public class Point
{
    private int x;        // x-coordinate of the point
    private int y;        // y-coordinate of the point

    public Point()
    {
        x = 0;
        y = 0;
    }

    public Point(int a, int b)
    {
        x = a;
        y = b;
    }

    // Other methods not shown
}
```

The following incomplete class declaration is intended to extend the above class so that points can be named.

```
public class NamedPoint extends Point
{
    private String name; // name of point

    // Constructors go here

    // Other methods not shown
}
```

Consider the following proposed constructors for this class.

I. 

```
public NamedPoint()  
{  
    name = "";  
}
```

~~II.~~

```
public NamedPoint(int d1, int d2, String pointName)  
{  
    x = d1;  
    y = d2;  
    name = pointName;  
}
```

~~III.~~

```
public NamedPoint(int d1, int d2, String pointName)  
{  
    super(d1, d2);  
    name = pointName;  
}
```

Which of these constructors would be legal for the `NamedPoint` class?

- ~~(A)~~ I only
- ~~(B)~~ II only
- (C) III only
- (D) I and III only
- ~~(E)~~ II and III only