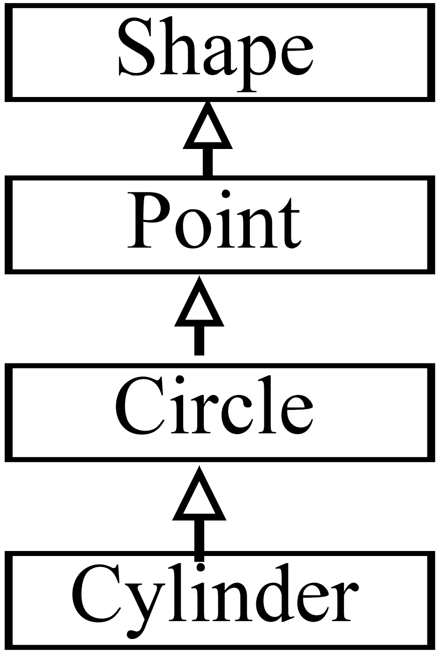
	getArea()	getVolume()	getName()	toString()
Shape	**0.0**	**0.0**	**null**	**default Object implementation**
Point	**0.0**	**0.0**	**“Point”**	**(x, y)**
Circle	**πr^2**	**0.0**	**“Circle”**	**Center=(x,y); Radius=r**
Cylinder	**$2\pi r^2+2\pi rh$**	**πr^2h**	**“Cylinder”**	**Center=(x,y); Radius=r; Height=h**



Shape hierarchy class diagram.

AppInher > PrInher > class Shape

```
public class Shape {
   // return area of shape; 0.0 by default
   public double getArea()
   {
      return 0.0;
   }

   // return volume of shape; 0.0 by default
   public double getVolume()
   {
      return 0.0;
   }

    public String getName() {
        return null;
    }
} // end class Shape
```

AppInher > PrInher > class Point

```
public class Point extends Shape {
  private int x;  // x part of coordinate pair
  private int y;  // y part of coordinate pair

  // no-argument constructor; x and y default to 0
  public Point()
  {
    // implicit call to Object constructor occurs here
  }

  // constructor
  public Point( int xValue, int yValue )
  {
    // implicit call to Object constructor occurs here
    x = xValue;  // no need for validation
    y = yValue;  // no need for validation
  }

  // set x in coordinate pair
  public void setX( int xValue )
  {
    x = xValue;  // no need for validation
  }

  // return x from coordinate pair
  public int getX()
  {
    return x;
  }

  // set y in coordinate pair
  public void setY( int yValue )
  {
    y = yValue;  // no need for validation
  }

  // return y from coordinate pair
  public int getY()
  {
    return y;
  }

  // override method getName to return "Point"
  public String getName()
  {
    return "Point";
  }

  // override toString to return String representation of Point
  public String toString()
  {
    return "(" + getX() + ", " + getY() + ")";
  }

} // end class Point
```

AppInher > PrInher > class Circle

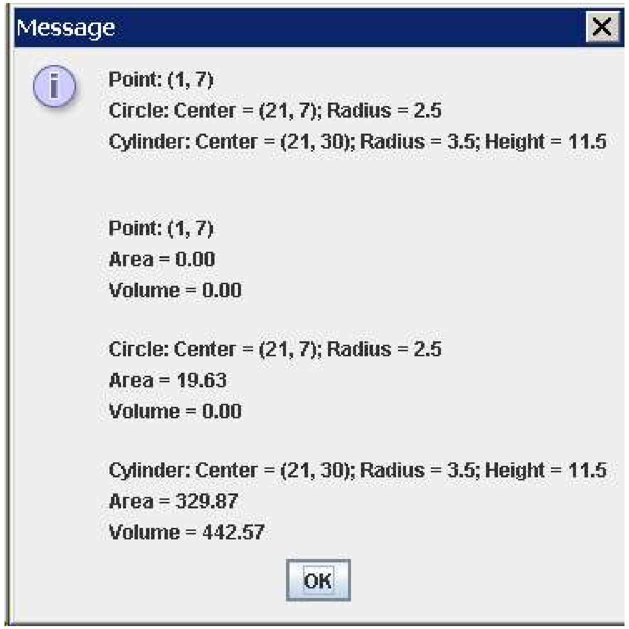
```
public class Circle extends Point {
  private double radius;  // Circle's radius

  // no-argument constructor; radius defaults to 0.0
  public Circle()
  {
    // implicit call to Point constructor occurs here
  }

  // constructor
  public Circle( int x, int y, double radiusValue )
  {
    super( x, y );  // call Point constructor
    setRadius( radiusValue );
  }

  // set radius
  public void setRadius( double radiusValue )
  {
    radius = ( radiusValue < 0.0 ? 0.0 : radiusValue );
  }

  // return radius
  public double getRadius()
  {
    return radius;
  }

  // calculate and return diameter
  public double getDiameter()
  {
    return 2 * getRadius();
  }

  // calculate and return circumference
  public double getCircumference()
  {
    return Math.PI * getDiameter();
  }

  // override abstract method getArea to return Circle area
  public double getArea()
  {
    return Math.PI * getRadius() * getRadius();
  }

  // override abstract method getName to return "Circle"
  public String getName()
  {
    return "Circle";
  }

  // override toString to return String representation of Circle
  public String toString()
  {
    return "Center = " + super.toString() + "; Radius = " + getRadius();
  }

} // end class Circle
```

AppInher > PrInher > class Cylinder

```
public class Cylinder extends Circle {
  private double height;  // Cylinder's height

  // no-argument constructor; height defaults to 0.0
  public Cylinder()
  {
    // implicit call to Circle constructor occurs here
  }

  // constructor
  public Cylinder( int x, int y, double radius, double heightValue )
  {
    super( x, y, radius );  // call Circle constructor
    setHeight( heightValue );
  }

  // set Cylinder's height
  public void setHeight( double heightValue )
  {
    height = ( heightValue < 0.0 ? 0.0 : heightValue );
  }

  // get Cylinder's height
  public double getHeight()
  {
    return height;
  }

  // override method getArea to return Cylinder area
  public double getArea()
  {
    return 2 * super.getArea() + getCircumference() * getHeight();
  }

  // override method getVolume to return Cylinder volume
  public double getVolume()
  {
    return super.getArea() * getHeight();
  }

  // override abstract method getName to return "Cylinder"
  public String getName()
  {
    return "Cylinder";
  }

  // override toString to return String representation of Cylinder
  public String toString()
  {
    return super.toString() + "; Height = " + getHeight();
  }

} // end class Cylinder
```

AppInher > PrInher > class TestInheritance

```
import java.text.DecimalFormat;
import javax.swing.JOptionPane;

public class TestInheritance {

  public static void main( String args[] )
  {
    // set floating-point number format
    DecimalFormat twoDigits = new DecimalFormat( "0.00" );

    // create Point, Circle and Cylinder objects
    Point point = new Point( 1, 7 );
    Circle circle = new Circle( 21, 7, 2.5 );
    Cylinder cylinder = new Cylinder( 21, 30, 3.5, 11.5 );

    // obtain name and string representation of each object
    String output = point.getName() + ": " + point + "\n" +
      circle.getName() + ": " + circle + "\n" +
      cylinder.getName() + ": " + cylinder + "\n";

    Shape arrayOfShapes[] = new Shape[ 3 ];  // create Shape array

    // arrayOfShapes[ 0 ] is a subclass Point object
    arrayOfShapes[ 0 ] = point;

    // arrayOfShapes[ 1 ] is a subclass Circle object
    arrayOfShapes[ 1 ] = circle;

    // arrayOfShapes[ 2 ] is a subclass Cylinder object
    arrayOfShapes[ 2 ] = cylinder;

    // loop through arrayOfShapes to get name, string
    // representation, area and volume of every Shape in array
    for ( int i = 0; i < arrayOfShapes.length; i++ ) {
      output += "\n\n" + arrayOfShapes[ i ].getName() + ": " +
        arrayOfShapes[ i ].toString() + "\nArea = " +
        twoDigits.format( arrayOfShapes[ i ].getArea() ) +
        "\nVolume = " +
        twoDigits.format( arrayOfShapes[ i ].getVolume() );
    }

    JOptionPane.showMessageDialog( null, output );  // display output

    System.exit( 0 );
  } // end main
} // end class TestInheritance
```



```
Point: (1, 7)
Circle: Center = (21, 7); Radius = 2.5
Cylinder: Center = (21, 30); Radius = 3.5; Height = 11.5


Point: (1, 7)
Area = 0.00
Volume = 0.00

Circle: Center = (21, 7); Radius = 2.5
Area = 19.63
Volume = 0.00

Cylinder: Center = (21, 30); Radius = 3.5; Height = 11.5
Area = 329.87
Volume = 442.57
```