public void draw(Graphics g, Rectangle area, ImageObserver view) {

float scale = getScale(area);

int y = area.y;

// Draw the title separately

SlideItem titleItem = new TextItem(0, getTitle());

Style titleStyle = Style.getStyle(titleItem.getLevel());

drawSlideItem(titleItem, area.x, y, scale, g, titleStyle, view);

y += titleItem.getBoundingBox(g, view, scale, titleStyle).height;

// Draw the remaining slide items

for (SlideItem slideItem : getSlideItems()) {

Style itemStyle = Style.getStyle(slideItem.getLevel());

drawSlideItem(slideItem, area.x, y, scale, g, itemStyle, view);

y += slideItem.getBoundingBox(g, view, scale, itemStyle).height;

}

}

private void drawSlideItem(SlideItem slideItem, int x, int y, float scale, Graphics g, Style style, ImageObserver view) {

slideItem.draw(x, y, scale, g, style, view);

}

The refactored code makes use of a private helper method called **drawSlideItem** to avoid duplicating the code for drawing each slide item. The **for** loop is also simplified by using a for-each loop instead of an index-based loop. The variables are given more descriptive names, and the comments are rephrased to provide more context. The overall structure of the code is also more clear and easier to understand.