```
/**
 * Main client class
 *
 * As a student at Union College, I am part of a community that
values intellectual effort, curiosity and discovery. I understand
that in order to truly claim my educational and academic
achievements, I am obligated to act with academic integrity.
Therefore, I affirm that I will carry out my academic endeavors
with full academic honesty, and I rely on my fellow students to
do the same.
 *
 * @author Blair Hagen
 * @version 4-14-2016
 */
import CSLib.DrawingBox;
import Second.Block;

public class Client {
    public static void main(String[] args) {

        DrawingBox mainDrawBox = new DrawingBox();

        Block defBlock = new Block();
        Block cusBlock = new Block(100 ,250);
        defBlock.display(mainDrawBox);
        cusBlock.display(mainDrawBox);

        defBlock.setPosition(300, 175);
        defBlock.display(mainDrawBox);

        cusBlock.setPosition(200, 400);
        cusBlock.display(mainDrawBox);

        defBlock.setPosition(700, 200);
        defBlock.setDimensions(defBlock.getWidth() * 2,
defBlock.getHeight(), defBlock.getDepth() * 2);
        defBlock.display(mainDrawBox);


        DrawingBox secondDrawBox = new DrawingBox();
        Block blockArray[] = new Block[5];

        for (int i = 1; i <= blockArray.length; i++)
        {
            blockArray[i - 1] = new Block(100, 100 * i);
            blockArray[i - 1].display(secondDrawBox);
        }
    }
}
```

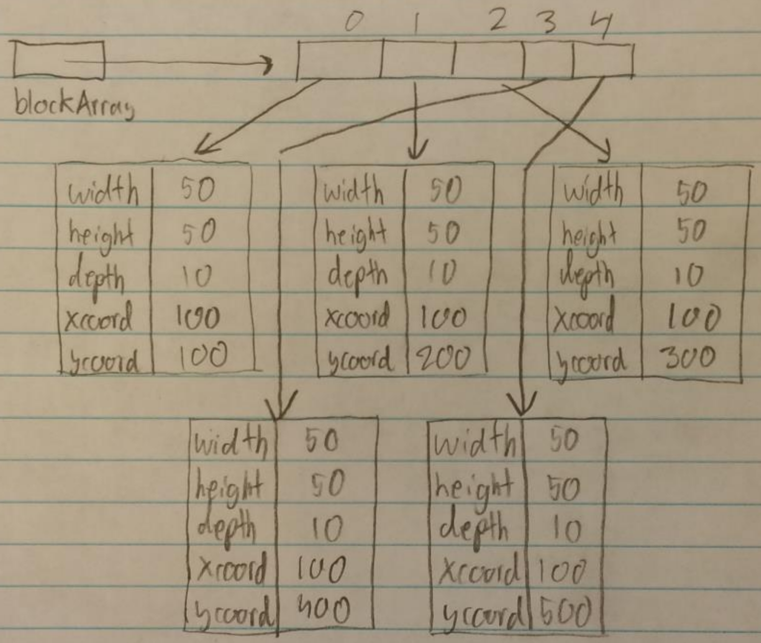
```java
package original;
/**
 * Represents three-dimensional blocks
 *
 * @author Blair Hagen
 * @version 4-14-2016
 *
 */
import CSLib.DrawingBox;


public class Block {

     private int width;
     private int height;
     private int depth;
     private int xcoord;
     private int ycoord;

     private static int DEFAULT_WIDTH = 50;
     private static int DEFAULT_HEIGHT = 50;
     private static int DEFAULT_DEPTH = 10;
     private static int DEFAULT_XCOORD = 100;
     private static int DEFAULT_YCOORD = 100;


     /**
      * Default constructor.
      * Creates block based on default values.
      */
     public Block() {

          width = DEFAULT_WIDTH;
          height = DEFAULT_HEIGHT;
          depth = DEFAULT_DEPTH;
          xcoord = DEFAULT_XCOORD;
          ycoord = DEFAULT_YCOORD;

     }

     /**
      * Creates block at inputted x/y coord with
      * default size.
      */
     public Block(int xcoord, int ycoord) {

          width = DEFAULT_WIDTH;
          height = DEFAULT_HEIGHT;
          depth = DEFAULT_DEPTH;
          this.xcoord = xcoord;
          this.ycoord = ycoord;

     }

     /**
      * Getter for box width
      *
      * @return width
      *         The width of the box
      */
     public int getWidth() {
          return(width);
     }

     /**
      * Getter for box height
      *
      * @return height
      *         The height of the box
      */
     public int getHeight() {
          return(height);
     }

     /**
      * Getter for box depth
      *
      * @return depth
      *         The depth of the box
      */
     public int getDepth() {
          return(depth);
     }

     /**
      * Getter for box x-coord
      *
      * @return xcoord
      *         The x-coordinate of the box
      */
     public int getXCoord() {
          return(xcoord);
     }

     /**
      * Getter for box y-coord
      *
      * @return ycoord
      *         The y-coordinate of the box
      */
     public int getYCoord() {
          return(ycoord);
```

```
    }


    /**
     * Setter for box coordinates
     *
     * @param xcoord
     *          Desired x-coordinate for box
     * @param ycoord
     *          Desired y-coordinate for box
     */
    public void setPosition(int xcoord, int ycoord) {
        this.xcoord = xcoord;
        this.ycoord = ycoord;
    }

    /**
     * Setter for box dimensions
     *
     * @param width
     *          Desired width for box
     * @param height
     *          Desired height for box
     * @param depth
     *          Desired depth for box
     */
    public void setDimensions(int width, int height, int depth)
{
        this.width = width;
        this.height = height;
        this.depth = depth;
    }


    /**
     * Displays box in a given drawingbox
     *
     * @param box
     *          Drawingbox to draw box in
     */
    public void display(DrawingBox box) {
        for (int i = 0; i < depth; i++)
        {
            box.drawRect(getXCoord() + i*2, getYCoord() +
i*2, getWidth(), getHeight());
        }

    }
}
```

```
package Second;

/**
 * This version of block class uses only
 * two instance variables to define a block.
 *
 * @author Blair Hagen
 * @version 4-14-2016
 */
import CSLib.DrawingBox;
import java.awt.Rectangle;

public class Block {

      private Rectangle rect;
      private int depth;

      private static int DEFAULT_WIDTH = 50;
      private static int DEFAULT_HEIGHT = 50;
      private static int DEFAULT_DEPTH = 10;
      private static int DEFAULT_XCOORD = 100;
      private static int DEFAULT_YCOORD = 100;

      /**
       * Default constructor
       * Creates block based on default values
       */
      public Block() {

            rect = new Rectangle(DEFAULT_XCOORD, DEFAULT_YCOORD,
DEFAULT_WIDTH, DEFAULT_HEIGHT);
            depth = DEFAULT_DEPTH;

      }

      /**
       * Creates block at inputted x/y coord with
       * default size
       */
      public Block(int xcoord, int ycoord) {
            rect = new Rectangle(xcoord, ycoord, DEFAULT_WIDTH,
DEFAULT_HEIGHT);
            depth = DEFAULT_DEPTH;
      }


      /**
       * Getter for box width
       *
       * @return width
       *         The width of the box
       */
      public int getWidth() {
            return((int)rect.getWidth());
      }

      /**
       * Getter for box height
       *
       * @return height
       *         The height of the box
       */
      public int getHeight() {
            return((int)rect.getHeight());
      }

      /**
       * Getter for box depth
       *
       * @return depth
       *         The depth of the box
       */
      public int getDepth() {
            return(depth);
      }

      /**
       * Getter for box x-coord
       *
       * @return xcoord
       *         The x-coordinate of the box
       */
      public int getXCoord() {
            return((int)rect.getX());
      }

      /**
       * Getter for box y-coord
       *
       * @return ycoord
       *         The y-coordinate of the box
       */
      public int getYCoord() {
            return((int)rect.getY());
      }


      /**
       * Setter for box coordinates
       *
       * @param xcoord
       *         Desired x-coordinate for box
```

```
     * @param ycoord
     *            Desired y-coordinate for box
     */
    public void setPosition(int xcoord, int ycoord) {
        rect.setLocation(xcoord, ycoord);
    }

    /**
     * Setter for box dimensions
     *
     * @param width
     *            Desired width for box
     * @param height
     *            Desired height for box
     * @param depth
     *            Desired depth for box
     */
    public void setDimensions(int width, int height, int depth)
{
        rect.setSize(width, height);
        this.depth = depth;
    }

    /**
     * Displays box in a given drawingbox
     *
     * @param box
     *            Drawingbox to draw box in
     */
    public void display(DrawingBox box) {
        for (int i = 0; i < depth; i++)
        {
            box.drawRect(getXCoord() + i*2, getYCoord() +
i*2, getWidth(), getHeight());
        }
    }
}
```

Box array memory diagram



width	50
height	50
depth	10
xcoord	100
ycoord	100

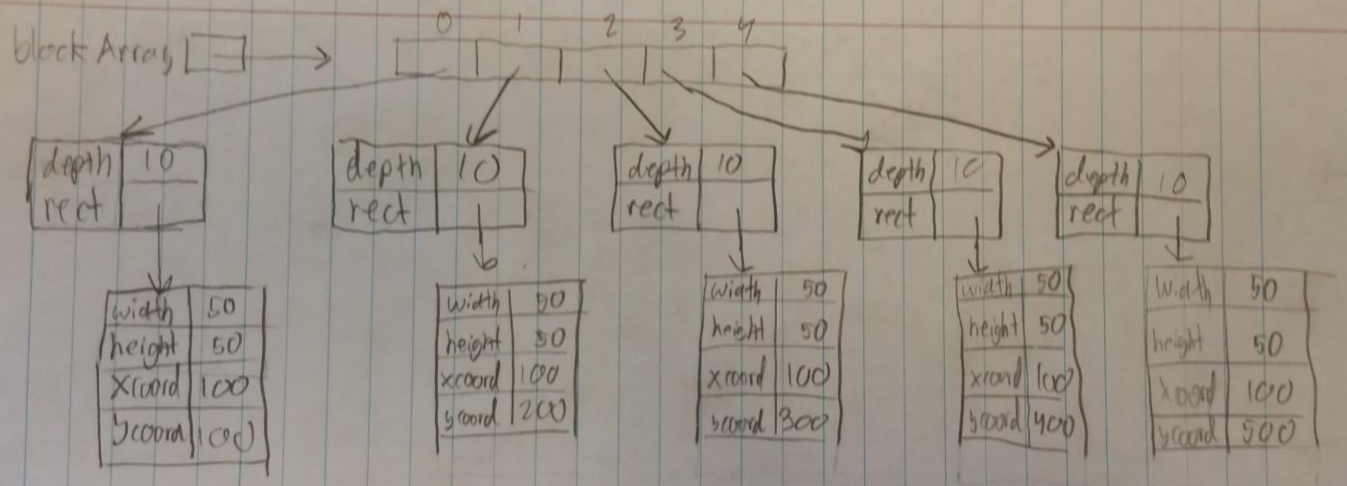
width	50
height	50
depth	10
xcoord	100
ycoord	200

width	50
height	50
depth	10
Xcoord	100
ycoord	300

width	50
height	50
depth	10
Xcoord	100
ycoord	400

width	50
height	50
depth	10
Xcoord	100
ycoord	500

Second array memory diagram on back ⟶


block Array
0
1
2
3
4
depth 10
rect
width 50
height 50
Xcoord 100
Ycoord 100
depth 10
rect
width 50
height 50
xcoord 100
ycoord 200
depth 10
rect
width 50
height 50
xcoord 100
ycoord 300
depth 10
rect
width 50
height 50
xcoord 100
ycoord 400
depth 10
rect
width 50
height 50
xcoord 100
ycoord 500