Introduction to Java – Objects English Name: ____________________

Worksheet: `BoundingBox` Class

Write `class BoudingBox` that has:

- Four private fields: `left`, `bottom`, `right`, and `top`, each of which stores a real number, and which together represents the bottom-left and top-right corners on a coordinate system.
- A constructor that takes four real parameters to set the four fields in the order given above.
- A getter for each of the fields (...but actually, just write one on your worksheet).
- An instance method named `width` that takes no parameters and returns the width of the `BoudingBox` object.
- An instance method named `height` that takes in no parameters and returns the height of the `BoudingBox` object.
- An instance method named `area` that returns the area of the `BoudingBox` object.
- An instance method named `encloses` that takes two parameters, `x` and `y`, which store real numbers and represents two coordinates in the coordinate system, and returns `true` if the coordinates are within the `BoudingBox` object, otherwise returns `false`.
- An instance method named `equals` that takes one `BoundingBox` object parameter and returns `true` if all four coordinates are the same in both objects, otherwise returns `false`.
- An instance method named toString that prints out the bounding box as two points on a coordinate system with one point being the bottom left and the other being the top right, for example: `(1.00, 1.00), (3.00, 5.00)`.

You do not need to perform error checking for any real value being “not a number” or “infinity”, however, the `BoundingBox` class must ensure `left` is more left than `right` on a horizontal number line and `bottom` is lower than the `top` on a vertical number line. Write the class in the space below and on the back of this page.

English Name: ____________________

Worksheet: BoundingBox Class

```java
public class BoundingBox {
   private double left, bottom, right, top;
   public BoundingBox(double left, double bottom,
                      double right, double top) {
      this.left = Math.min(left, right);
      this.right = Math.max(left, right);
      this.bottom = Math.min(bottom, top);
      this.top = Math.max(bottom, top);
   }
   public double width() {
      return right-left;
   }
   public double height() {
      return top-bottom;
   }
   public double area() {
      return this.width() * this.height();
   }
   public boolean encloses(double x, double y) {
      return left <= x && x <= right &&
           bottom <= y && y <= top;
   }
   public boolean equals(BoundingBox b) {
      return this.left == b.left &&
             this.right == b.right &&
             this.bottom == b.bottom &&
             this.top == b.top;
   }
   @Override
   public String toString() {
      return String.format("(%.2f, %.2f), (%.2f, %.2f)",
                           left, bottom, right, top);
   }
   public double getLeft() {
      return left;
   }
   public double getRight() {
      return right;
   }
   public double getBottom() {
      return bottom;
   }
   public double getTop() {
      return top;
   }
}
```