**Question 2 (10 marks total):**
Consider the following `Range` class where a `Range` object represents a range of integer values between some minimum and maximum value:

```
public class Range implements Comparable<Range> {
    // class invariant: this.width >= 0
    private int min;
    private int width;

    // remainder of class not shown
}
```

A range having zero width includes only the minimum value of the range. A range includes all of the integer values from `this.min` up to and including `this.min + this.width`. For example, a range with a minimum value of 5 and a width of 3 includes the values 5, 6, 7, and 8.

Two `Range` objects are equal if their minimum values are equal and their widths are equal.

**Part A (3 marks):**

Provide an implementation of the `compareTo` method where the minimum values of the two ranges `r1` and `r2` are compared such that:

- `r1` is less than `r2` if the minimum of `r1` is less than the minimum of `r2`
- `r1` is greater than `r2` if the minimum of `r1` is greater than the minimum of `r2`
- `r1` is "equal" to `r2` if the minimums of the two ranges are equal

**Part B (3 marks):**

A `Range` object overlaps another `Range` object if the two objects have at least one value in common in their ranges of values.

Suppose that a student implements `compareTo` such that:

- `r1` is "equal" to `r2` if the two ranges overlap, otherwise
- `r1` is less than `r2` if the maximum of `r1` is less than the minimum of `r2`, otherwise
- `r1` is greater than `r2` if the minimum of `r1` is greater than the maximum of `r2`

Explain whether such an implementation of `compareTo` is consistent with `equals`.

**Part C (4 marks):**

Provide an implementation of a non-static method `overlaps(Range other)` that returns `true` if this range overlaps the `other` range, and `false` otherwise.