CMPUT 350 Lab 4 Prep Problems

---

1. For class

```
struct Point {
    int x{};
    int y{};
};
```

in `p1.cpp`, implement global operators `<<` and `>>` that allows you to read and write points from `stdin`/`stdout` like so:

```
Point p;
std::cin >> p;
std::cout << p << std::endl;
```

Note: For this to work, you need to include `iostream`.

---

2. In file `p2.cpp`, implement the following class operators for class `Point` above:

`==` `>` `>=`

which work componentwise. I.e., two points are equal if their `x` components match and their `y` components match. For `>` and `>=`, use the lexicographic ordering with `x` being the higher-valued component.

---

3. In file `p3.cpp`, implement `pre++`, `post++`, `pre--`, `post--`, for class `Point` above. Their effect is to increment or decrement a point's `x` component, respectively. Also, test your implementation.

---

4. Suppose you want to count how many `Point` instances have been constructed at any given time when your program runs. Write code in `p4.cpp` that gives you access to this information. Can your solution be compromised by a teammate working on a different project file? If so, try to improve your design.