Operator overloading

Operator overloading allows the developer to define the meanings of a built-in operator.

From homework #1:

```
friend std::ostream &operator<<(std::ostream &out, const Points2D &
                                                    some_points)

friend std::istream &operator>>(std::istream &in, Points2D &some_points)
```

Let's work on:

```
friend std::istream &operator>>(std::istream &in, Points2D &some_points){
```

What do we wan to do?

```
//de allocate any sequence in some_points

if some_points.sequence_ is NOT NULL
      then deallocate
      and set some_points.sequence_ to NULL

//get the size
      in>> somepoints.size_;

//we create the sequence
  some_points.sequence_;

//populate sequence
      for i to some_points.size_:
        for j to 2 {
          //set point i jth coordinate from the stream;
```

We would have to do error handling

```
        in >> same_points.sequema[i][j];
    Return in;
}
```