EECE.3220 Spring 2020: Exam 1
Class Definitions for Section 4 & Extra Credit

`Point` definition

```
class Point {
private:
   double xCoord, yCoord;
public:
   Point();
   Point(double x, double y);
   double getX();
   double getY();
   void setX(double x);
   void setY(double y);
   void printPoint(ostream& out);
};
```

Selected `Point` functions are defined below. The “get” and “set” function definitions for this class are not shown, but (hopefully) their operation should be self-explanatory.

```
Point::Point() : xCoord(0), yCoord(0) {}

Point::Point(double x, double y) : xCoord(x), yCoord(y) {}

void Point::printPoint(ostream& out) {
   out << "(" << xCoord << ", " << yCoord << ")";
}
```

The `PointList` class is defined on the other side of this sheet.

EECE.3220 Spring 2020: Exam 1
Class Definitions for Section 4 & Extra Credit

PointList definition
Notes:

```
class PointList {
private:
   Point list[50];   // A PointList can hold up to 50 points
   unsigned np;      // Actual number of points stored in array
public:
   PointList();                          // Default constructor
   PointList(vector <Point> pts);   // EXTRA CREDIT
   void addPoint(Point& p);          // Add a single point
                                         //   at the end of the list
   void printList(ostream& out);    // QUESTION 4A
   bool isFull();                        // QUESTION 4B
   bool isLine();                        // QUESTION 4C
};
```

Selected PointList function definitions are shown below:

```
PointList::PointList() : np(0) {}

void PointList::addPoint(Point& p) {
   if (np < 50)
      list[np++] = p;
   else
      cout << "Error: list full\n";
}
```