|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  |  | [ [Home](http://people.uncw.edu/cferner/index.html) ][ [Classes](http://people.uncw.edu/cferner/Classes/ClassesMain.html) ][ [Research](http://people.uncw.edu/cferner/Research/ResearchMain.html) ][ [Links](http://people.uncw.edu/cferner/LinksMain.html) ][ [Biography](http://people.uncw.edu/cferner/BioMain.html) ] |  |
| [Up](http://people.uncw.edu/cferner/Classes/csc434/Assignments.html)  [CS Dept.](http://www.uncw.edu/csc/)     [http://people.uncw.edu/cferner/Classes/csc434/glogo50.gif](http://www.uncw.edu/) |  | **CSC 434**  **Assignment 2 (Perl) - Due 2/28/2017 ~~2/21/2017~~**  Write a Perl program to determine if two line segments intersect.  You should create a couple of classes to do this.  Create a class called Point. It should have an X and Y that can be in the ranges 0 <= X < MAX\_X and 0 <= Y < MAX\_Y.  Use constants for MAX\_X and MAX\_Y.  The class should have the following functions (with appropriate parameters and return values):   1. A constructor 2. getX 3. getY 4. setX 5. setY 6. setXY 7. random - assigns random values to X and Y within their appropriate ranges   Create a class called LineSegment.  A line segment should consists of two points: A and B. The class should have the following functions (with appropriate parameters and return values):   1. A constructor 2. getA 3. getB 4. setA 5. setB 6. random - assigns random values to the two points A and B   Create a function that takes two line segments as parameters and determines if they intersect.  For example, in the figure below, (a) and (b) intersect, but (c) and (d) do not. Be careful to handle line segments that are parallel or nearly parallel. You may also need to take special care with lines that are nearly vertical, depending on how you solve the problem.  http://people.uncw.edu/cferner/Classes/csc434/assign1_fig.jpeg  In the main program, create an array of many pairs (at least 20) of line segments with random points.  Determine which ones intersect.   Perl is installed on **webdev** and comes with most Linux installations.  You can use the interpreter just by type the command **perl**.  If you put your Perl program in a file, you can run that by typing **perl <file>**.  There is a Perl plugin for Eclipse, called EPIC.  On my version of Eclipse (NEON 2), I had to use the test version of EPIC as the normal distribution doesn't work.  The test version of EPIC can be found at  <http://www.epic-ide.org/updates/testing>.You may also find this site (<http://www.epic-ide.org/running_perl_scripts_within_eclipse/entry.htm>) to be helpful in getting a Perl program to run in Eclipse.  **Documentation:**  It is expected that you will follow standard practices of documentation of your program.  That means that classes and methods should have header information include: Author, date written, list and description of parameters or data members (where appropriate), type and description of return values (where appropriate), a general description of the purpose of the class or method.  You should also put in comments for code for which its meaning is not obvious.  That does not mean to put in comments that a loop will loop through some values, but rather put in comments for things that it would take study or searching documentation to understand.  You should also put in comments for constructs that are not common in every language (e.g. use of regular expressions).  And finally, any part of the program that asks the user for input should be proceeded with an appropriate prompt.  An appropriate prompt is one that tells the user what to enter and in what form.  For example, "Enter two floating-point numbers separated by whitespace on one line."  **Deliverables:**   * The source file(s) * Explain the features you used in the program. In particular, explain what the language constructs, libraries, and functions do.  You do not need to explain things that are in most languages such as **for** loops.  But if you use some utility, like say a regular expression, then explain what it is and how it works. * Submit the files to the assignment drop box on BlackBoard.  You can get there by login into UNCW MySeaPort, then clicking on the link for the course. |  |
| This page was last updated: **February 28, 2017**  **Email: http://people.uncw.edu/cferner/images/mailme.png**  [ [Home](http://people.uncw.edu/cferner/index.html) ][ [Classes](http://people.uncw.edu/cferner/Classes/ClassesMain.html) ][ [Research](http://people.uncw.edu/cferner/Research/ResearchMain.html) ][ [Links](http://people.uncw.edu/cferner/LinksMain.html) ][ [Biography](http://people.uncw.edu/cferner/BioMain.html) ] | | |  |