|  |  |
| --- | --- |
| **Application/ Program name:** | L2-1 |
| **Written by:** | Bailey Nichols |

|  |
| --- |
| **Purpose or problem definition:** |
| This program would have been written to demonstrate the use of overloading operators to make a program that measured rectangles by their area, and allowed the user to add and subtract rectangles by area.  I spent like 5 hours fixing the class you gave us and after that I just moved on to other things. |
|  |
| **Program Procedures:** |
|  |
|  |
| **Algorithm/Processing/Conditions:** |
| **Inputs:** |
|  |
| **Processes:** |
|  |
| **Outputs:** |
|  |
|  |
| **Notes & Restriction:** |
| A.Overload the pre– and post– increment and decrement operators to increment and decrement, respectively, the length and width of a rectangle by one unit. (Note that after decrementing the length and width, they must be positive.)  B.Overload the binary operator - to subtract the dimensions of one rectangle from the corresponding dimensions of another rectangle. If the resulting dimensions are not positive, output an appropriate message and do not perform the operation.  C.The operators == and != are overloaded by considering the lengths and widths of rectangles. Redefine the functions to overload the relational operator by considering the areas of rectangles as follows: two rectangles are the same, if they have the same area; otherwise, the rectangles are not the same. Similarly, rectangleyard1 is greater than rectangleyard2 if the area of yard1 is greater than the area of yard2. Overload the remaining relational operators using similar definitions.  D.Write the definitions of the functions to overload the operators defined in parts A through C.  E.Write a test program that tests various operations on the class rectangleType. |
|  |
| **Comments:** |
| code on the handout sheet that does not compile |