SCSSE, University of Wollongong
CSCI204/MCS9204 Object and Generic Programming in C++
Laboratory Exercise Week 3

Task 1: Debugging

Download two programs **Worker.cpp** and **Office.cpp** from the subject web site.

Each of these programs contains various errors; syntax and/or logical. Copy the files to your own working directory and fix the dugs so the programs can be compiled and run properly.

Task 2: Basic class/object construction

Create a program taxreturn.cpp. It should contain a class TaxReturn with the following:

- Data fields holding a tax ID number, last name, first name, annual income, number of dependents and the amount of tax owed.
- Constant static fields holding the tax rates in accordance with the following array:

Income	0 dependents	1 dependent	2+ dependents
0 – 10,000	0.10	0.08	0.07
10,001 – 30,000	0.12	0.11	0.09
30,001 – 60,000	0.18	0.15	0.13
60,001+	0.25	0.22	0.19

- A static member function that displays the tax table.
- A member function `setAll()` to set all the field values on the basis of the tax ID, last and first names, annual income and number of dependents.
- A member function `calculateTax()` to determine the tax owed.
- A member function `display()` to display all the information for a taxpayer.

In the main() function:
- declare three objects of the class TaxReturn for three taxpayers.
- implement a loop to read the necessary details from the standard input stream for three taxpayers. Then display tax owed by each taxpayer.

In writing the code you should consider carefully whether data fields and functions should be public or private. Defining data members use appropriate data types and meaningful names.

Task 3: Object and class diagrams

Download the file **Diagram.pdf** from the subject web site.

Look at the object diagram `Polygon` and draw the associated class diagram, taking into account and explaining the following considerations:

1. What are the appropriate multiplicities?

2. Each point has an x coordinate and a y coordinate.

3. What is the smallest number of points required to construct a polygon?

4. Does it matter whether points can be shared between polygons?

5. We need to take into account that the points need to be ordered in the description. We haven't come across this yet but we can place a constraint (ordered) into the diagram as a label opposite (across the association) the multiplicity.