ESS201 – C++ Programming
International Institute of Information Technology Bangalore

Miniproject: FileModifier
Announcement on Nov 05, 2020 (Wednesday), at 3:30 pm IST
Submission by Nov 20, 2020 (Friday), at 11:59 pm IST, on Domjudge
Demo in the week of Nov 23, 2020, schedule to be announced.

Highlight:
This minproject has to be implemented with as many concepts as possible that you have learned in the C++ module.

You will be implementing an application that reads in a file, modifies it, and writes it out in the same or a related format.

Your score will be proportional to the usage of concepts such as inheritance, polymorphism, and composition, and the usage of the standard template library. You will get a score for the group effort in integrating the task combination for your group. The project can be integreated by implementing an abstract base class, which can based on the file format call an appropriate derived class. Implementation and testing of your code independently as well as in the integrated mode are expected.

Find the task descriptions below with descriptions, resources with examples and expected outputs. The output must be a file.

General instructions of implementation:

- All the files are to be read in this assignment must use ASCII file format.
- Write your own functions to read the file using std::string. Do not attempt to use third-party file-reading API or codes written by others. Your code must reflect your understanding of the file format.
- The executable for the project must be able to take the filename as a command-line argument for the executable.
- "Read the file" implies reading a sample file, along with testing the same out. Since these are text files, you can test your code even with files you create using the file format specifications. However, it must be ensured that the files you create are readable on applications that can load such file formats, e.g. GIMP should be able to read the .ppm file you create.

General instructions on group activity:

- Each student will work on a designated task, and each group will have a unique combination of tasks.
- Assign a group leader for your team, and the group leader submits for the team.
  - In case you would like for us to look at your individual contributions separately, make an individual submission. This is *not* needed if your contribution in the group submission is exactly the same as what you would like to make as an individual submission.
  - The group leader can submit the individual one, as need be, along with the group one.
  - Each submission must be a tarred zipped file (*.tar.gz) of the folder containing code divided into folders for header files (named as "include") and source files (named as "src").

Task1: GDF file reader:
https://gephi.org/users/supported-graph-formats/gdf-format/
Use a file for weighted graph, which could be directed or undirected.

- Read the file.
- Create an adjacency matrix.
- Output the adjacency matrix in a .csv format.

Task2: Image file reader:
https://www.cs.swarthmore.edu/~soni/cs35/f13/Labs/extras/01/ppm_info.html
https://people.sc.fsu.edu/~jburkardt/data/ppma/ppma.html

- Read the .ppm file.
- Change the color of each pixel from RGB to BRG format.
- Output a .ppm file which is viewable on a GIMP application.

Task3: CSV file reader:

- Read the file.
  - You can use any .csv file, e.g. UCI ML repository has several files: https://archive.ics.uci.edu/ml/index.php
- Store the data from the file as a table/matrix, including the row and column field names, if any, and transpose it.
- Output the transposed table in .csv format.

Task4: .txt file reader:

- Read the file, as big as books.
  - You can use any text file, e.g. Project Gutenberg has several classics in .txt format: https://www.gutenberg.org/ebooks/
- Generate the document statistics, as provided in gedit text editor application.
- Write the document statistics as a header and the current file as body, and generate a new .txt file.
- Output the new .txt file.

Task5: UCINET DL file reader for graphs:
https://gephi.org/users/supported-graph-formats/ucinet-dl-format/
Use a file with "Edge list", labels, and weighted edges.

- Read the file.
- Sort the vertices/nodes based on its degree, and rank it based on the descending order of degree.
  - Degree of a node is computed as the sum of the weights of all the edges sharing the node.
  - Re-index the nodes based on its rank.
  - Filter the graph nodes to contain only the top 50% of the nodes in the descending order, and the edges which are between the filtered nodes.
- Output the filtered graph in the UCINET DL format.

Task6: PLY file reader for surface meshes:
https://people.math.sc.edu/Burkardt/data/ply/ply.html

- Read the file.
- Compute the area of each face in the graphical object, and sort the faces by area.
    - Each face is a triangle, hence, the formula for area of the triangle can be used.
    - For polygon with more than 3 vertices, you can pick one vertex and introduce edges from the vertex to all other vertices, in addition to its existing edges. The new edges “subdivide” the polygon to triangles. The area of the polygon is the sum total of the areas of the new triangles.
    - Discard faces which are in the lower 10% of the sorted list of triangles.
    - Prune vertices which are now not shared by any faces.
- Output the filtered surface mesh in a .ply file format.

Task7: Image file reader:
https://people.sc.fsu.edu/~jburkardt/data/ppma/ppma.html
https://people.sc.fsu.edu/~jburkardt/data/pgma/pgma.html

- Read a .ppm file.
- Convert the color image to grayscale by using the formula for converting color value of a pixel to grayscale:
    - Gr = ((0.3 * R) + (0.59 * G) + (0.11 * B)).
- Output the image in .pgm file, which is viewable on a GIMP application.

Task8: CSV file reader:

- Read the file.
- Store the data from the file as a table/matrix, including the row and column field names, if any, and sort it in ascending order using two different columns, as is available in Excel.
    - The column names, as used in Excel (A, B, ..., AA, AB, ...) can be entered as commandline arguments along with file name.
- Output the sorted table in .csv format.
