1. Frame rates at varying window sizes

- **1x1:** ~114–118 FPS
- **300x300:** ~115–119 FPS
- **Fullscreen:** ~118–120 FPS

Explanation:
The above frame rates are very consistent for different window sizes. There lies very little variation in the frame rates. This consistency in frame rate can be explained through several reasons:

1. Vertical Sync (V-Sync): V-Sync has been activated for my system, thereby locking the frame rate to the refresh rate of the monitor.
2. GPU Capability: It is expected that the graphics card might be powerful enough to handle the rendering of this relatively simple scene at a rate higher than the one at which the monitor could refresh, especially for larger window sizes.
3. Unchanging Geometry: The real geometry rendered, that is, gears, does not change with the increase of the window size. This slight increase in frame rates for larger sizes can be reasoned by reduced overhead in handling very small window sizes.
4. Fill Rate Efficiency: Modern GPUs are optimized for large draw calls, which might explain the minor performance increase with increased window size.

2. Variation of frame rate between small round number and thousands of FPS
The following can explain the differences in frame rates for small round numbers and very high numbers:

1. V-Sync: If on, it limits the frame rate to the refresh rate of the monitor; for example, 60, 72, or 85 FPS.
2. Uncapped Frame Rates: With V-Sync disabled, the GPU can render frames as quickly as possible, and for simple scenes, this can correspond to several thousands of FPS.
3. Hardware Differences: Faster GPUs can render simple scenes very quickly, if not shackled by V-Sync or other limiting factors.
4. Driver Settings: Certain graphics drivers can define the limit of frame rate or other optimizations which provide performance clues.
5. API Overhead: The efficiency of the utilized graphics API herein-OpenGL-and the way it is implemented will impact the highest achievable frame rate.
6. CPU Bottleneck: In a few cases, particularly for very high frame rates, the CPU may become a bottleneck in preparing and issuing draw calls to the GPU.

3. Time Estimate This lab, if done by hand, would typically take about thirty to sixty minutes, including the following:
Downloading and installation of the required libraries- 20 minutes
Run the program for different window sizes – 10 minutes
Research to explain findings – 90 minutes
Write up the answers - 30 minutes
Total time: 150 minutes