Checklist (red = ignore, green = ask, highlight = include)

Project Software (30%)		
Checklist	**Points**	**Grade**
Compilable	1	
Runtime errors	1	
Readbility	2	
Math / Technical Knowledge from other areas	2	
OS independent	1	
IDE independent	1	
Backward / Forward Compatibility	1	
Code Organization	2	
Efficiency	2	
Robustness	1	
Memory Leaks	1	
Memory Usages	1	
CPU Usages	1	
Complexity and Creativity	2	
Write and include your own libraries	1	
All of the following are subjected to proper usages.		
All Primitive Data Types	0.5	
Global and Static variables	0.5	
Operators	0.5	
Conditional Statements	0.5	
Iterative Statements	0.5	
Functions	0.5	
Recursions	0.5	
Static functions	1	
Preprocessor	1	
Pointers	1	
References	1	
Function Parameter Passing	0.5	
Function Overloading	0.5	
Function Pointers	0.5	
Lambda Functions	0.5	
Template Functions	0.5	
Templated Lambda Functions	0.5	
Variadic Templates	0.5	
Containers	2	
Iterators	1	
Enum	1	
Union	1	
Struct / Class	2	
Objects	2	
Template Class	1	
Operator Overloading	1	
Inheritance	2	
Virtual Function / Overriding	1	
GUI	1	
Date & Time	0.5	
File System Handling	0.5	
Mouse & Keyboard	0.5	
Regular Expression	0.5	
Concurrent Programming	1	
Socket / Communication	1	
Total	50	

Douri: after I get the C++ plot statically (entire orbit)

Rowan: Matlab to C++

In coding and software development, "backward compatibility" and "forward compatibility" are important concepts related to how software or systems handle changes and updates:

- **Backward Compatibility (Backward Compat)**:
- Backward compatibility, also known as "backward compat," refers to the ability of a system, software, or a new version of a product to work with data or components that were created or designed for an older version.
- In other words, if a system is backward compatible, it can accept and process data or interact with components created using a previous version without issues.
- Backward compatibility is essential to ensure that users can upgrade to newer versions of software or hardware without breaking their existing data or configurations.
- **Forward Compatibility (Forward Compat)**:
- Forward compatibility, also known as "forward compat," is the ability of a system, software, or a new version to work with data or components that are designed for a future version that may not yet exist.
- In essence, a forward-compatible system can process data or interact with components that were created with features or formats that are expected to be introduced in future versions.
- Forward compatibility is less common and more challenging to implement because it requires anticipating and accommodating potential future changes and updates.

In summary:

- **Backward compatibility** is about ensuring that newer versions are compatible with older data or components, allowing for smooth transitions and upgrades.
- **Forward compatibility** is about ensuring that older versions can work with data or components that might be created in the future, which is a more complex and less common requirement.

Both types of compatibility are important in software and system design, and achieving the right balance between them is essential to maintain a good user experience and minimize disruptions during software upgrades or changes.

-
- In C++, a **static variable** is a variable that is associated with a class or function rather than with instances of that class or function

- **Recursion** is a programming or mathematical concept where a function calls itself in order to solve a problem. In the context of programming, a recursive

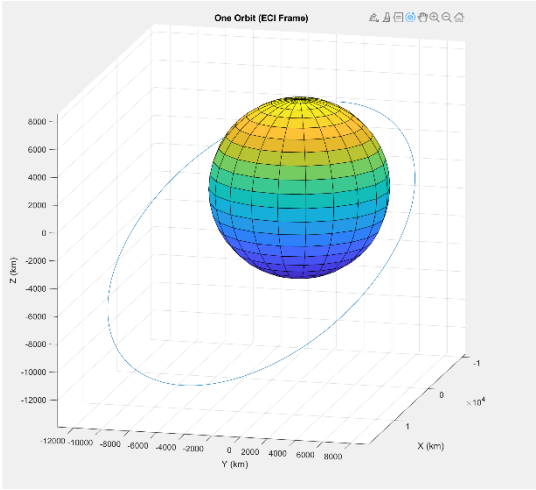
function is a function that breaks down a problem into smaller, similar subproblems and solves them iteratively until it reaches a base case. Recursion is a powerful and elegant technique, especially for solving problems that can be naturally divided into smaller instances of the same problem.

- the **preprocessor** is a component of the compilation process that performs text manipulation on the source code before it's actually compiled.

enum (short for enumeration) is a data type that consists of a set of named integer constants

Tasks:

- Bring matlab code to c++ (Rowan)



- Plot orbit statically(Douri)
- Math tools (.h file) function:
    - Dot product Douri
    - Matrix template
    - **Plotting in 3D (function?) test this for any arbitrary** Douri
        - Input: x,y. output: “plotting”
        - Test plotting a quadratic function
        - Adding titles to the plot, legend
    - Newton rhapson