

FLORIDA INTERNATIONAL UNIVERSITY



Knight Foundation School of Computing and Information Sciences

Senior Project Final Presentation – Spring 2024

Geometric Visualization Tool

Team Members: Alex Castaneira, Anthony Thompson, Jinpan Zheng, Alex Narvaez, Joseph Mejia

Product Owner: Antonio Hernandez

Professor: Masoud Sadjadi

Problem Description

- Geometric visualization tool is a capstone project in which a developed tool has been made primarily filling the direct needs for students taking Computational Geometry at FIU.
- The general purpose of the Geometric visualization tool is to serve students, teachers, and developers in working on geometric structures visualizations using java graphics or programs.
- The project tries to achieve this purpose by focusing on the implementation of basic geometric objects, basic geometric data structures, and basic geometric algorithms that can be used for graphing using whatever java compatible IDE.
- Our visualization tool addresses the current shortage of geometric algorithms libraries in Java. The tool helps fill this gap to help provide more efficient development for visualization of geometric data structures and algorithms

Problem Management

- The tool sets itself apart by being an abstract framework utilizing to its full capabilities two integral external libraries needed, the CGAL library and a java private JAR library called JGeom.
- With this the tool functionalities currently revolve around aiding in the design, implementation, and testing of geometric data structures that a student would struggle with in java programs.
- With CGAL's integration it's robust capabilities with Java's strengths in scholastic setting the task of visualizing geometric objects, becomes easier for developers, students, and teachers by becoming a powerful resource for quicker designing, implementing, and testing geometric algorithms.
- This integration enhances functionality and access by circumvention of CGAL library necessity for data in C++ through txt files, meaning now only necessary for 1 group member in software development to have component.

Main Focus

This capstone semester was focused on:

1. Building out the initial abstract framework with goal of adding graphing collections to java programs.
2. Develop with simplicity scaling in mind for the general geometric objects, geometric data structures, geometric algorithms.
3. Complete and sound system architecture built with system having a loop design of imports and exports capabilities both sides.

User Stories

Java Codebase:

- The abstract framework built has the attribute of a simple method or function for adding graphing collections of geometric objects to the java program.
- Formatting of and standardization of toString objects lines and rays that are not simple shapes, that later create or are the building blocks for Voronoi diagram and Delaunay triangulation into the text file.
- Create simple parser in java for the appropriate text files in which information derived from C++ toString is printed in desired format in java program for values such as red color, blue color, green color, boundary, transparency, x-coordinate, y-coordinate, point, vertices.
- Implement export and import aspects for java side of the codebase, in the java program only has import aspects and complete export now. With these aspects users can specify or test only specific objects instead of all geometric objects.

C++ Codebase:

- Utilizes computational geometry capabilities from CGAL to visualize geometric objects
- Create simple program exercises that can implement the two geometric objects of doubly connected edge list (DCEL) and Voronoi diagram in a bare bones method that all can use.
- Reformat and break down the existing C++ project codebase side developed, regarding toString implementation and text file saving, into appropriate and standardized C++ architecture with a .h file, .cpp file, and main file for appropriate requested direction.
- Implement export and import aspects for C++ side of the codebase which only has export and complete import now. With these aspects users can specify or test only specific objects instead of all geometric objects.

Technologies Used



CGAL: The computational geometry algorithms library which makes possible the software project by giving reliable access to geometric algorithms and data structures as a C++ library.



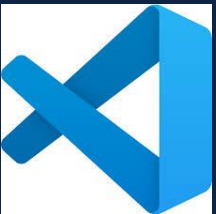
IntelliJ: The IDE user for our java codebase in which development, testing, and graphing was done. It is also the IDE we used for appropriate JGeom library.



C++: The language that was used to write half our code for the abstract framework specifically files `main.cpp`, `geo2util.cpp` and `geo2util.h` plus the testing for export side.

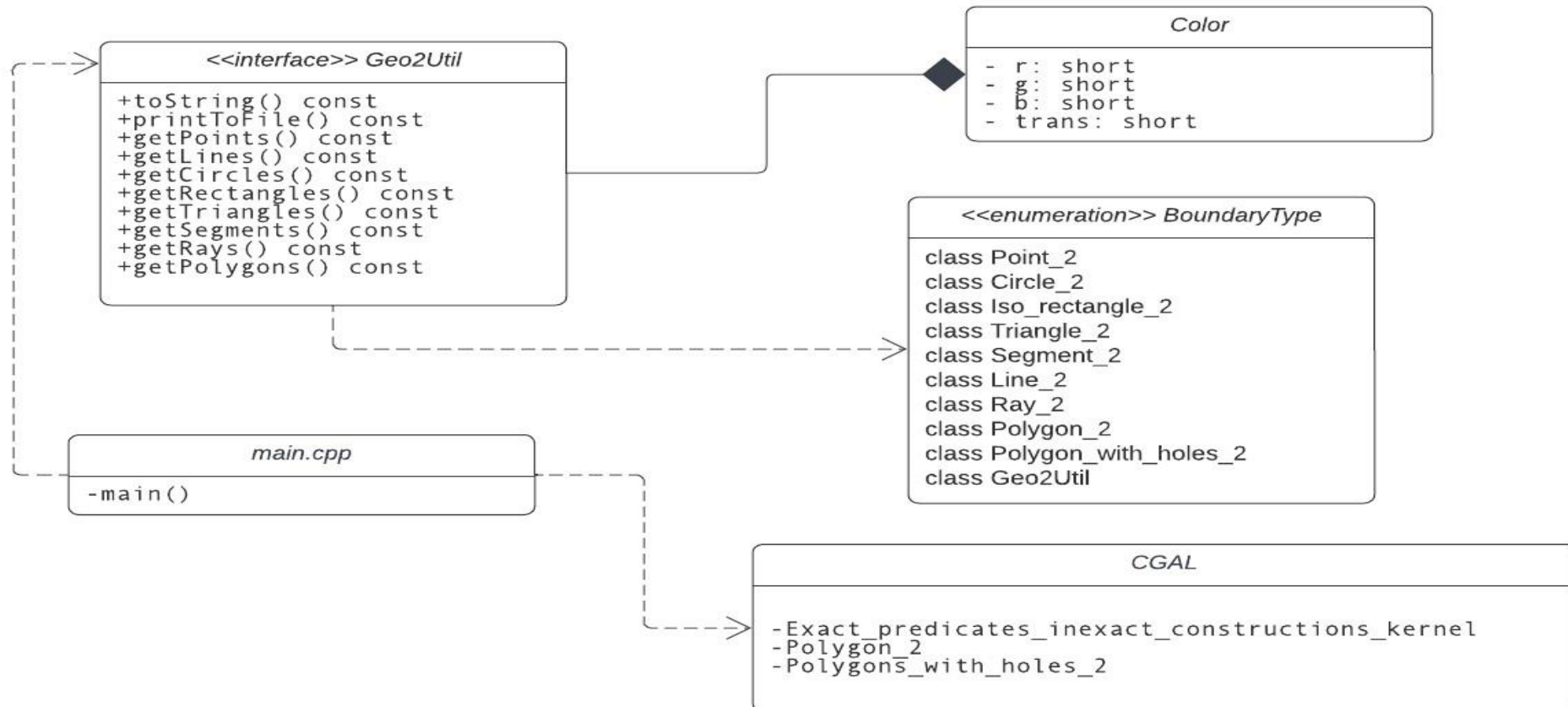


JAVA 21: The language that was used to write half of our code for our abstract framework tool, Geometric Visualization Tool, specifically files `FileUtil`, `TestFileRead`.

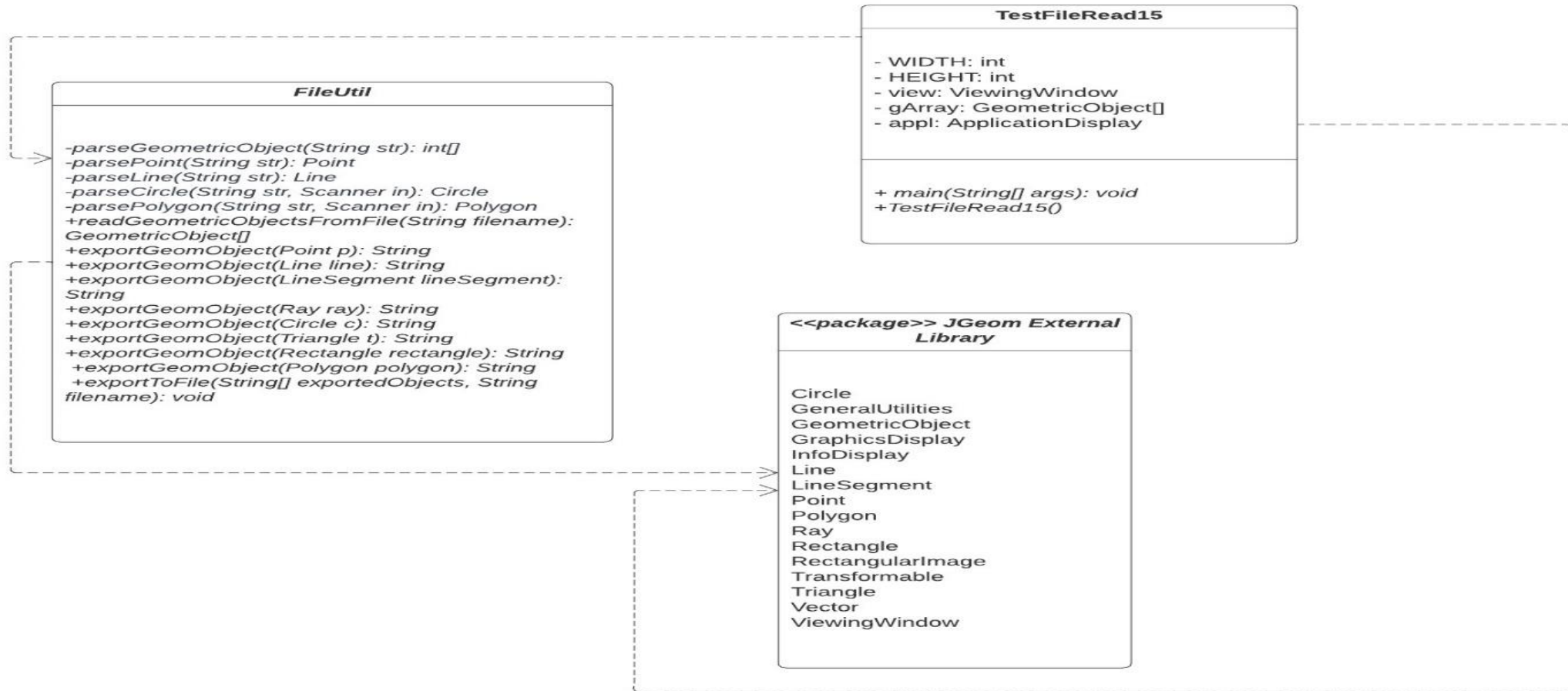


Visual Studio Code: The IDE and Code Editor for our C++ code development with relevant C++ libraries utilized.

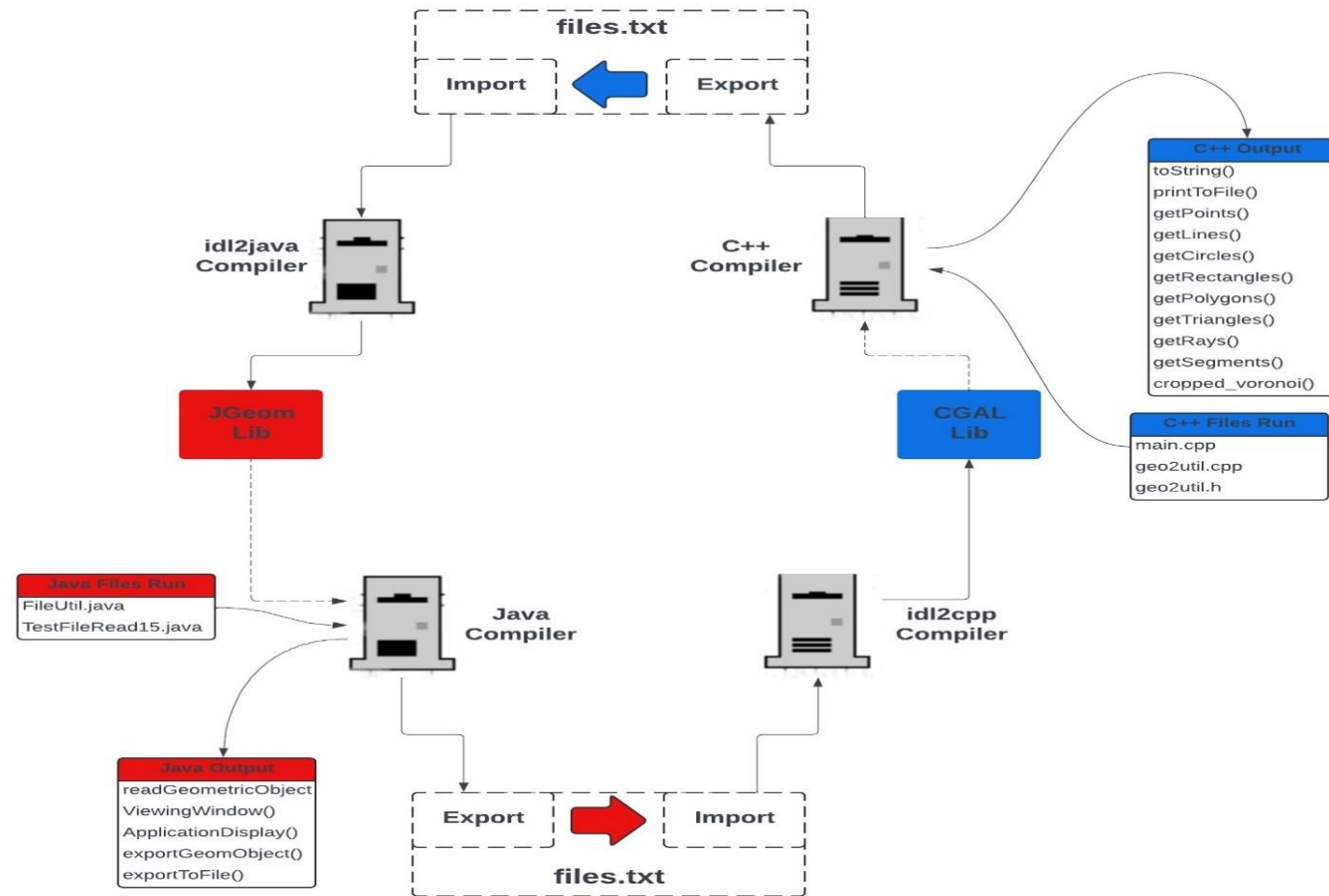
Object Design Diagram



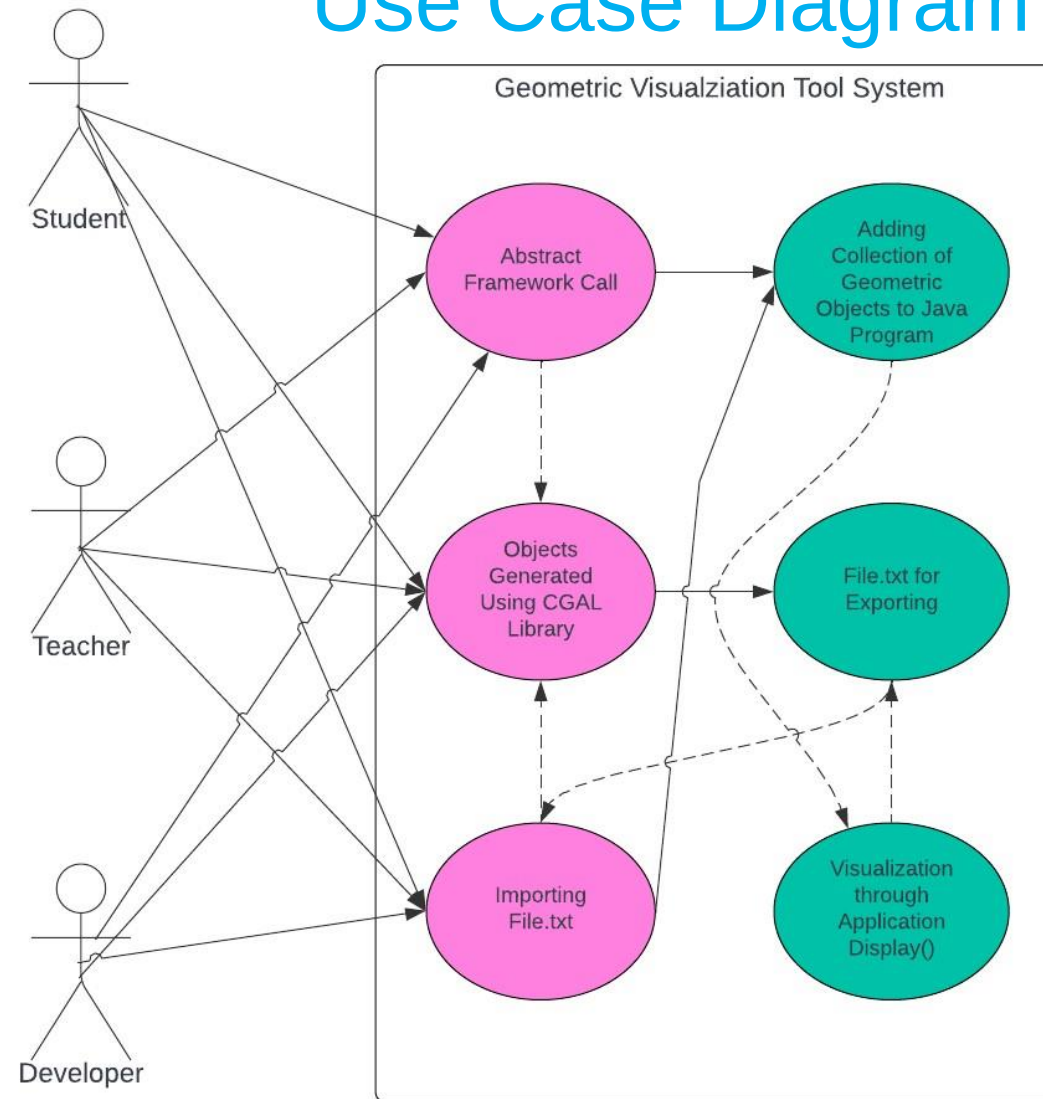
Object Design Diagram



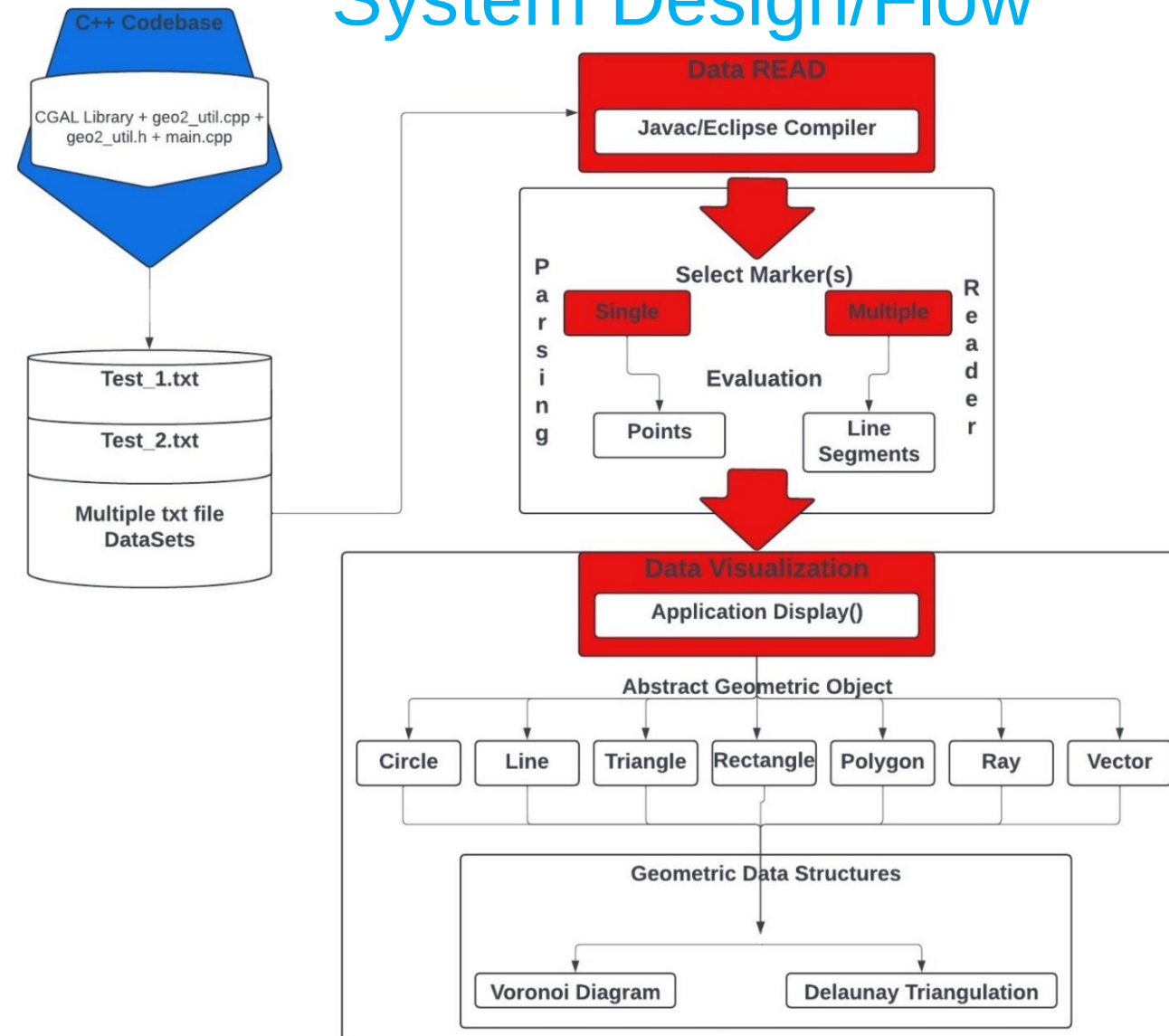
System Architecture



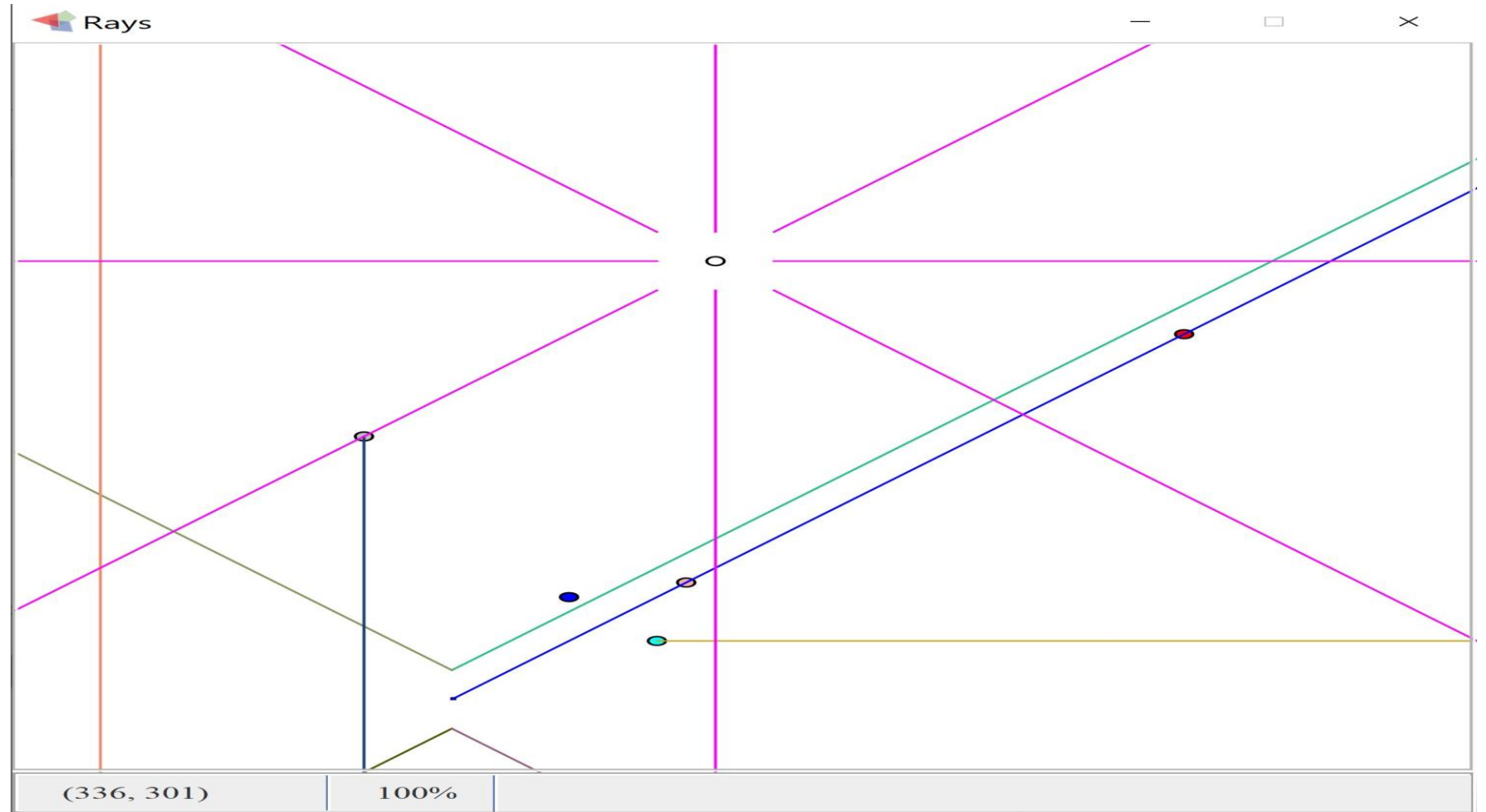
Use Case Diagram



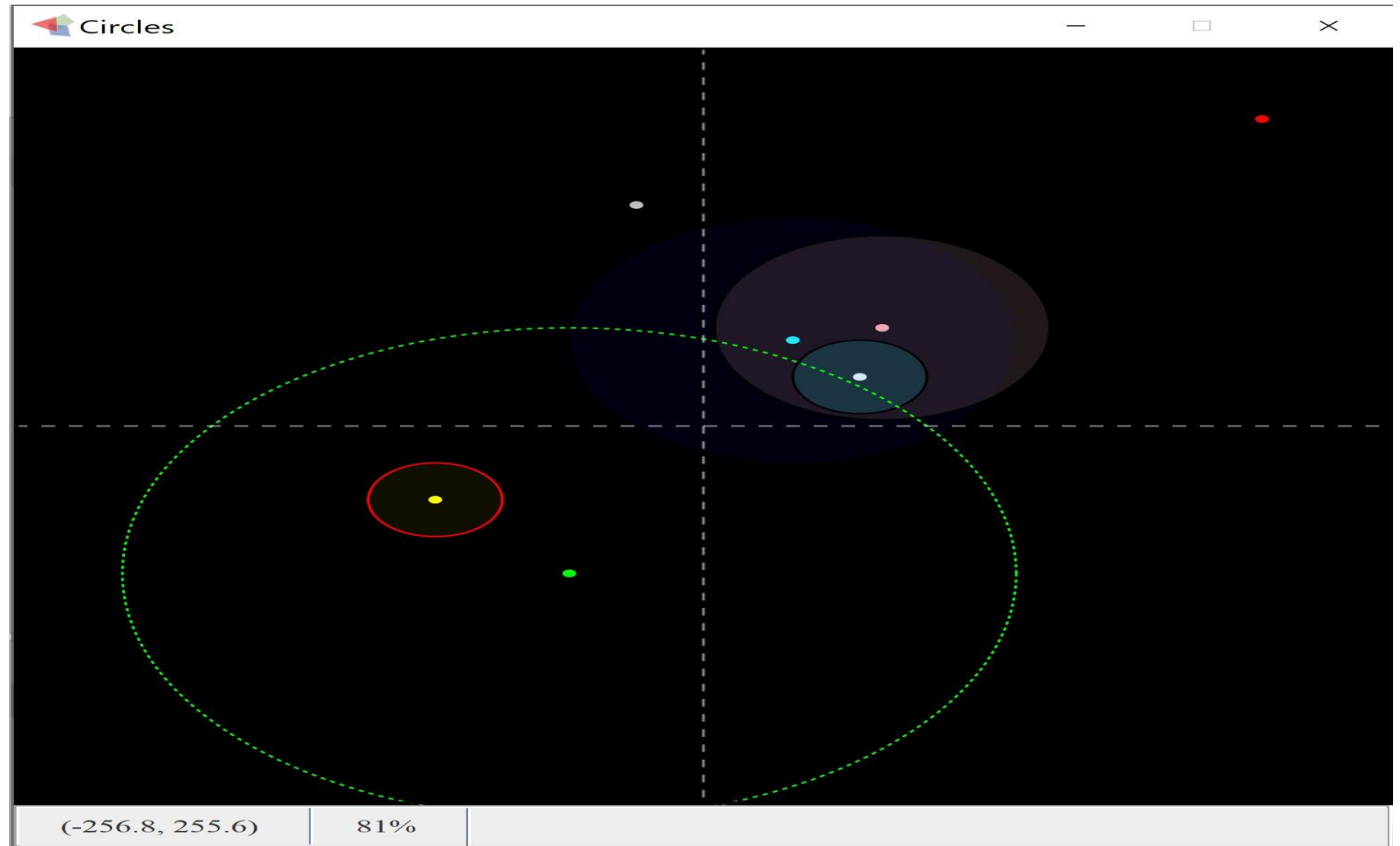
System Design/Flow



Rays Output Window



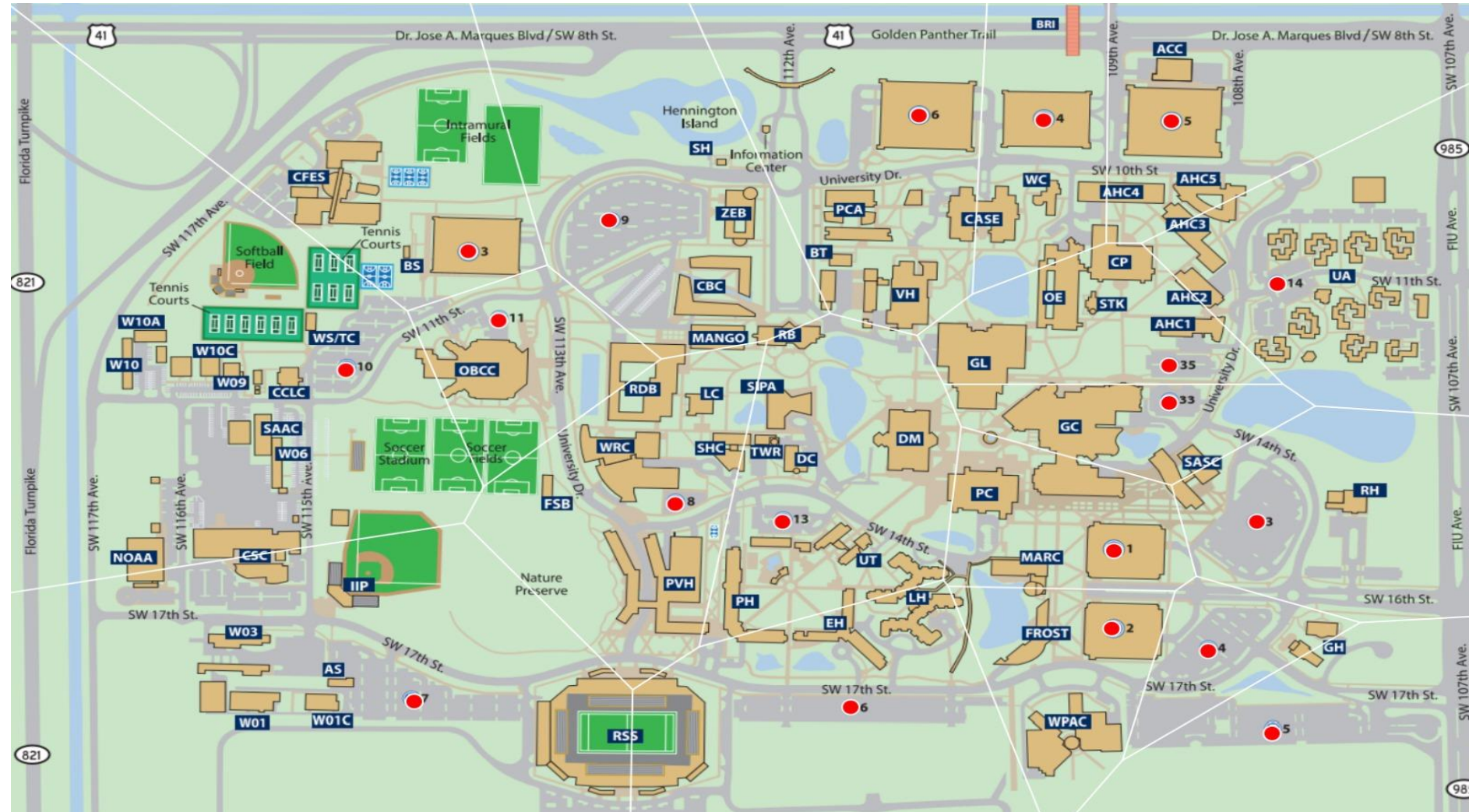
Circles Output Window



FIU



FIU



Contact Info

- Alex Castaneira – acast529@fiu.edu
- Anthony Thompson – athom351@fiu.edu
- Jinpan Zheng – jzhen022@fiu.edu
- Alex Narvaez – anarv014@fiu.edu
- Joseph Mejia – jmeji118@fiu.edu



Summary

- In this project, the Geometric Visualization Tool gave us the opportunity to fully learn computational geometry without the background and to develop a software educational tool. We from the ground up built the documentation, tester classes, architecture in C++, the development of diagrams, the implementation of exporting and importing for two different components, C++ part and Java part of abstract framework. We all extended or refined our knowledge regarding Java graphics beyond the basics and CGAL capabilities.
- With the help of our product owner Mr. Hernandez, we were given both freedom and guidance in the developmental phase by creating our own testing classes to fully learn the components and capabilities this tool has right now. With Jinpan's leadership in C++ development phase and Anthony's leadership in Java importing development phase we were able to have a product early on that could be proudly showcased. We seek further improvements by next capstone group regarding system architecture design left off by Alex C. We are also excited to see the complexity of geometric data structures and algorithms scale up in the future.



References

- Board, C. E. (n.d.). *The Computational Geometry Algorithms Library*. Sitewide ATOM. <https://www.cgal.org/>
- Graphics (java platform SE 8). (2024, January 8). <https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html>
- Hernandez, A. (2024). *JGeomLibrary, a Java Framework to Visualize Geometric Algorithms*. [Unpublished manuscript].
- Netscape Internet Service Broker for Java: Programmer's guide, version 1.0. (n.d.). <https://docs.oracle.com/cd/E19957-01/816-5698-10/basics.htm>