The Data.h header file defines a C++ class called Data designed to handle interest rate data. It includes essential C++ Standard Library headers, such as <map> and <string>, to provide the necessary functionality. The class offers three public member functions: a constructor, loadFromFile for loading data from a specified file, and getInterestRate for retrieving interest rates based on maturity time. Additionally, the getData function grants read-only access to an internal std::map containing maturity times as keys and their corresponding interest rates as values. This header file's use of header guards ensures that its contents are included only once in any given source file, preventing duplicate inclusion and potential errors.

The Data class constructor initializes the object.
The loadFromFile function attempts to open an Excel file specified by filePath. If the file is successfully opened, it reads each line, extracts maturity time and interest rate values, and stores them in the data map.
The getInterestRate function takes a maturityTime parameter and searches the data map for the corresponding interest rate. If found, it returns the interest rate; otherwise, it returns 0.0.
The getData function provides read-only access to the entire data map by returning a constant reference to it.
The Bond.h header file encapsulates the definition of the C++ class Bond, representing a financial bond. It includes header guards to ensure single inclusion during compilation and includes the necessary headers for input/output operations (<iostream>) and string manipulation (<string>).
Within the class definition, private member variables are declared to store information about the bond, including its expiration date, frequency of coupon payments, and coupon rate. Public member functions, such as constructors, a destructor, and methods for calculating bond prices and generating string representations, are provided to interact with and manipulate bond objects.
The Data class constructor initializes the object.
The loadFromFile function attempts to open an Excel file specified by filePath. If the file is successfully opened, it reads each line, extracts maturity time and interest rate values, and stores them in the data map.
The getInterestRate function takes a maturityTime parameter and searches the data map for the corresponding interest rate. If found, it returns the interest rate; otherwise, it returns 0.0.
The getData function provides read-only access to the entire data map by returning a constant reference to it.

The Bond.h header file defines a C++ class called Bond that represents a financial bond. This class encapsulates important bond attributes, including the expiration date, frequency of payments, and coupon rate, as private member variables. The header file includes necessary C++ Standard Library headers, such as <iostream> and <string>, to facilitate input and string manipulation. The Bond class provides several constructors, including a default constructor, a parameterized constructor for initializing bond properties, and a copy constructor for creating copies of bond objects. Additionally, a destructor is implemented to ensure proper resource cleanup.

The public interface of the Bond class offers two key functions: ToString, which returns a string representation of the bond's attributes, and PriceBond, which calculates the bond's price based on provided parameters like time to maturity, payment frequency, coupon rate, and interest rate. Header guards (#ifndef, #define, #endif) are employed to prevent multiple inclusions of this header file in the same translation unit, thereby avoiding duplication and potential compilation errors.

The Bond.cpp source file implements the functionalities of the Bond class defined in the corresponding Bond.h header file. It provides a default constructor and a parameterized constructor for creating Bond objects with specific attributes, along with a copy constructor and an empty destructor. Additionally, it offers a ToString method that returns a formatted string representation of a Bond object and a PriceBond function for calculating the price of a bond based on time to maturity, payment frequency, coupon rate, and interest rate.

The Test.h header file defines a C++ class named Test that appears to be intended for testing functionalities related to financial bonds, possibly involving the Bond class. This class includes the necessary headers, such as <unordered_map> and "Bond.h", for testing and accessing the required functionality. Its public interface consists of a default constructor and a runTests method, indicating its purpose to execute a series of test cases. Additionally, the class has private member functions, test1 and test2, which are likely individual test cases. Furthermore, an instance of the Bond class named a is included within the class. The use of header guards (#ifndef, #define, #endif) ensures that this header file is included only once in any given translation unit, preventing duplication and potential compilation errors.

The test.cpp source file defines a C++ class named Test responsible for conducting two specific tests. In its constructor, it initializes a Bond object with predetermined bond attributes. The runTests method first reads data from a CSV file to populate an unordered map, subsequently calling two test functions. test1 assesses data retrieval accuracy from the map by asserting the equality of specific keys with expected values. test2 examines the bond pricing functionality by calculating a bond's price using the PriceBond method and comparing it to an expected value within a specific precision. The file utilizes standard C++ libraries, such as <iostream>, <cassert>, <unordered_map>, <cmath>, and <iomanip>, while including the "Bond.h" header to access the Bond class. It employs cout for test output and assert for assertion checks, facilitating validation of the Bond class's behavior through comprehensive testing.