Exercise Description:

You are required to design a system that computes statistical measures (average, maximum, and minimum) for objects that can be "measured." These objects could be instances of classes like `BankAccount` and `Student`, which implement a `Measurable` interface.

Details:

1. **Measurable Interface (Measurable.java)**:
   - This is an interface that any class must implement to be "measurable."
   - It contains one method, `getMeasure()`, which returns a double value representing the measurable quantity (e.g., balance for a bank account, GPA for a student).
2. **BankAccount Class (BankAccount.java)**:
   - Implements the `Measurable` interface.
   - It contains a `balance` attribute, representing the balance of a bank account.
   - Implements the `getMeasure()` method to return the balance of the account.
3. **Student Class (Student.java)**:
   - Implements the `Measurable` interface.
   - It contains a `gpa` attribute, representing the student's GPA.
   - Implements the `getMeasure()` method to return the GPA.
4. **Statistics Class (Statistics.java)**:
   - This class is used to collect and compute statistics (average, maximum, and minimum) on objects that implement the `Measurable` interface.
   - It has attributes to keep track of the count of items, sum of the measures, as well as the current maximum and minimum.
   - Methods include:
     - `add(Measurable item)`: Adds a measurable object to the statistics, updating the sum, maximum, and minimum.
     - `getAverage()`: Returns the average of all added measurable objects.
     - `getMaximum()`: Returns the object with the highest measure.
     - `getMinimum()`: Returns the object with the lowest measure.
5. **Main Class (Main.java)**:
   - The `Main` class creates instances of `BankAccount` and uses the `Statistics` class to compute the average, maximum, and minimum balances.
   - Example:
     - Three bank accounts are created with different balances.
     - These bank accounts are added to the `Statistics` object.
     - The program prints the average, maximum, and minimum balance using the `Statistics` class.

Task:

1. Implement the `Measurable` interface, `BankAccount` class, `Student` class, and the `Statistics` class.

2. Create a `Main` class that demonstrates how the `Statistics` class can be used to compute statistics for different objects (e.g., BankAccount, Student).
3. Extend the functionality by adding `Student` objects to the statistics and calculating their average, maximum, and minimum GPA.

Bonus:

- Implement additional methods in the `Statistics` class for further insights (e.g., median).
- Modify the `Main` class to demonstrate how both bank accounts and students can be processed together in a single `Statistics` object.

This exercise focuses on object-oriented principles such as interfaces, classes, and statistical computation, making use of abstraction and polymorphism.