Exercise Description:

You are tasked with developing a Java application that allows users to input details about multiple products and determine which product has the best score-to-price ratio.

Details:

1. **Product Class (Product.java)**:
   - This class represents a product with attributes:
     - `name`: The name of the product (String).
     - `price`: The price of the product (double).
     - `score`: The score of the product (int).
   - **Constructor**:
     - The default constructor initializes the product with default values (empty name, price of 1, and score of 0).
   - **Methods**:
     - `read()`: This method reads product details from the user input using a `Scanner` and sets the `name`, `price`, and `score` of the product.
     - `printData()`: This method prints the product details (name, price, and score).
     - `is_better_than(Product other)`: This method compares the score-to-price ratio of the current product with another product and returns `true` if the current product has a better ratio, otherwise returns `false`.
2. **Main Class (Main.java)**:
   - The main method interacts with the user to continuously input details for multiple products.
   - **Process**:
     - The user is prompted to input multiple products.
     - Each product's details (name, price, and score) are entered, and the program compares the product with the best product found so far using the `is_better_than()` method.
     - The user can choose to enter more products or stop.
   - After all products are entered, the program prints the details of the product with the best score-to-price ratio.

Task:

1. Implement the `Product` class with attributes for name, price, and score, along with methods to read input, print data, and compare products based on their score-to-price ratio.
2. Implement the `Main` class to handle multiple products and determine the best product using the loop and comparison logic.
3. Allow the user to enter multiple products and stop when they choose to, then display the best product at the end.

Bonus:

- Modify the comparison logic to account for cases where two products have the same score-to-price ratio.
- Add validation to ensure that no negative or zero values are entered for price and score.

This exercise focuses on basic object-oriented programming concepts such as classes, methods, user input handling, and comparison logic in Java.