WIA1002 DATA STRUCTURE
LAB TEST 2
Duration: 1 hour
(Wednesday 11.00am – 12.00pm)

Question:
Imagine you are working on a project for a library management system. You need to implement data structures to manage the books that are borrowed and returned. You decide to use a stack for tracking borrowed books and a queue for managing book returns.

Requirements**:**

1. **Stack Implementation:** You are tasked with implementing a stack to track the books that are borrowed by library members.
    - Create a Java class named **BorrowedBooksStack**.
    - Implement the following functionalities:
        - **borrowBook(bookID)**: Pushes the **bookID** onto the stack when a book is borrowed.

          (1 mark)
        - **returnBook()**: Pops and returns the **bookID** from the stack when a book is returned.

          (1 mark)
        - **peekAtTopBook()**: Returns the **bookID** at the top of the stack without removing it.

          (1 mark)
        - **isEmpty()**: Returns true if the stack is empty, otherwise false.
        - **isFull()**: Returns true if the stack is full, otherwise false.

          (1 mark)

2. **Queue Implementation:** You also need to implement a queue to manage the returned books.
    - Create a Java class named **ReturnedBooksQueue**.
    - Implement the following functionalities:
        - **returnBook(bookID)**: Enqueues the **bookID** into the queue when a book is returned.

          (1 mark)
        - **processReturnedBooks()**: Dequeues and processes the returned books from the queue.

          (1 mark)
        - **peekAtFrontBook()**: Returns the **bookID** at the front of the queue without removing it.

          (1 mark)

- **isEmpty()**: Returns true if the queue is empty, otherwise false.
- **isFull()**: Returns true if the queue is full, otherwise false.

(1 mark)

3. **Main Method:** Write a tester program to test your implementation.
    1. Create objects for **BorrowedBooksStack** and **ReturnedBooksQueue**.
    2. Simulate **5** books borrowing using **borrowBook(bookID)** method.
    3. Check the top book ID using **peekAtTopBook()** method.
    4. Simulate book returning using **returnBook(bookID)** method.
    5. Process returned books using **processReturnedBooks()** method.
    6. Run the program and observe the output to verify functionality.

(2 mark)