Exercise Description:

You are tasked with creating a simple Java application with a graphical user interface (GUI) that allows the user to create a `Student` object by inputting a student's name and ID through text fields. The GUI will display the created student in the console after clicking a button.

Details:

1. **Student Class (Student.java)**:
    - Represents a student with attributes:
        - `name`: The student's name.
        - `id`: The student's ID.
    - **Constructor**:
        - Initializes the student's name and ID based on user input.
    - **Methods**:
        - `getName()` and `getId()`: Return the name and ID of the student.
        - `toString()`: Returns a string representation of the student, combining the name and ID.
2. **GUI Class (GUI.java)**:
    - This class extends `JFrame` to create the graphical interface.
    - **Components**:
        - `JTextField nameField`: A text field for inputting the student's name.
        - `JTextField idField`: A text field for inputting the student's ID.
        - `JButton createStudentButton`: A button that, when clicked, creates a `Student` object based on the input.
    - **Process**:
        - The `ButtonListener` inner class listens for button clicks and creates a `Student` object using the text entered in the fields.
        - The created student's details are printed to the console using the `toString()` method of the `Student` class.
3. **Main Class (Main.java)**:
    - The main class simply launches the GUI by creating an instance of the `GUI` class.

Task:

1. Implement the `Student` class with appropriate attributes and methods to handle student creation.
2. Implement the `GUI` class with text fields for user input and a button that creates a `Student` object based on the input.
3. In the `Main` class, launch the GUI and ensure the student details are printed to the console when the button is clicked.

Bonus:

- Add input validation to ensure that both the name and ID fields are filled out before allowing the student to be created.
- Display the student details directly in the GUI (e.g., in a label) instead of printing to the console.

This exercise focuses on building a simple GUI using Java Swing, handling user input, and using objects to manage student data.