Assignment: Encapsulation, Inheritance & Exception Handling

Instructions:

- Complete the following tasks using Python.
- Write well-structured code with comments explaining each section.
- Test your code to ensure it runs without errors.
- Submit your solutions as screenshots of the `.py` file or a Jupyter Notebook.

Task 1: Implementing Inheritance

Create a class `Vehicle` with the following attributes and methods:

- `brand` (string)
- `model` (string)
- `year` (integer)
- Method `display_info()` that prints the brand, model, and year.

Then, create a subclass `Car` that inherits from `Vehicle` and has an additional attribute:

- `fuel_type` (string)
- Override `display_info()` to include fuel type information.

Example Usage:

```
car1 = Car("Toyota", "Corolla", 2020, "Petrol")
car1.display_info()
```

Expected Output:

```
Brand: Toyota, Model: Corolla, Year: 2020, Fuel Type: Petrol
```

Task 2: Using Getters and Setters

Create a class `BankAccount` with:

- `account_number` (integer)
- `balance` (float)
- Getter method `get_balance()` to return the balance.
- Setter method `set_balance()` that prevents setting a negative balance and raises an exception if attempted.

Example Usage:

```
acc = BankAccount(12345, 1000)
print(acc.get_balance())
acc.set_balance(500)
print(acc.get_balance())
acc.set_balance(-200)  # Should raise an exception
```

Task 3: Implementing Try & Except

Modify the `BankAccount` class from Task 2 to handle exceptions properly:

- Catch invalid balance inputs using `try-except`.
- Display appropriate error messages instead of stopping execution.

Example:

```
try:
    acc.set_balance(-500)  # Should print an error message instead of
stopping
except Exception as e:
    print("Error:", e)
```

Task 4: Combining Concepts

Create a `Library` system using:

- **Inheritance:** Base class `LibraryItem` (attributes: `title`, `author`)
- **Subclass Book** (additional attribute: `isbn`)
- **Try & Except:** Handle errors when adding/removing books
- **Getter & Setter:** Use them for setting and retrieving book details

Example Usage:

```
book1 = Book("Harry Potter", "J.K. Rowling", "123456789")
print(book1.get_title())
```

Submission:

- Submit screenshots of the `.py` or `.ipynb` file containing your solutions.
- Ensure all tasks are completed and tested.
- Write comments explaining your approach in each section.

Good luck! 🚀