Task 1: Animal Hierarchy Create an abstract class called **Animal** that represents an animal. The **Animal** class should have pure virtual functions **sound()** and **diet()**. Then, create two subclasses **Dog** and **Cat** that inherit from the **Animal** class. Implement the **sound()** function in each subclass to print the sound made by a dog and a cat, respectively. Implement the **diet()** function to print the diet of a dog and a cat.

Task 2: Banking System Create an abstract class called **BankAccount** to represent a generic bank account. The **BankAccount** class should have the following pure virtual functions:

1. **void deposit(double amount)**: A function to deposit money into the account.
2. **void withdraw(double amount)**: A function to withdraw money from the account.
3. **void displayBalance()**: A function to display the current balance in the account.

Then, create two subclasses **SavingsAccount** and **CheckingAccount** that inherit from the **BankAccount** class. Implement the functions in each subclass according to their specific behaviors:

* For a **SavingsAccount**, the balance cannot go below zero, so the **withdraw** function should only allow withdrawals if the balance is greater than or equal to the withdrawal amount. Additionally, the account should accrue an interest of 2% on the balance after every deposit.
* For a **CheckingAccount**, there is no restriction on withdrawals, and there is no interest on the balance.

Task: Generic Stack Class Template

Create a generic Stack class template that can store elements of any data type. The Stack should support standard stack operations like **push**, **pop**, **top**, and **isEmpty**.