****

**Programming Fundamentals - Assignment 02**

**Task 1 (Code Already Provided): Design a Zoo Management System Using OOP**

**Objective**: The goal of this assignment is to design a simple Zoo Management System using the principles of Object-Oriented Programming that you learned in today's lecture. This will help you understand and apply the concepts of classes, objects, properties, and methods.

**Task Details**:

Your task is to create classes and objects for different types of animals in the zoo. Each animal has a few common attributes like name, age, and species, but they may also have unique behaviors.

**Instructions**:

1. **Animal Class**: Create a base class 'Animal' with the following attributes: name, age, and species. This class should have a method 'Eat()' that simply prints a message saying that the animal is eating.
2. **Specific Animal Classes**: Create subclasses for at least three types of animals (for example, Lion, Elephant, and Monkey). Each of these classes should inherit from the 'Animal' class.
   * Each specific animal class should override the 'Eat()' method to provide a unique message. For example, the 'Eat()' method in the 'Elephant' class might print a message saying "The elephant is eating peanuts."
   * Each specific animal class should have at least one unique method. For example, the 'Lion' class might have a method 'Roar()' that prints a message saying "The lion is roaring."
3. **Zoo Class**: Create a class 'Zoo' that has a list of Animals. This class should have methods to 'AddAnimal()' and 'FeedAllAnimals()'. The 'FeedAllAnimals()' method should call the 'Eat()' method on each animal in the zoo.
4. **Main Program**: In your main program, create an instance of the 'Zoo' class. Create instances of your specific animal classes and add them to the zoo. Then call the 'FeedAllAnimals()' method to show that each type of animal eats in its own way.

The code for the above assignment is already given. Your goal is to understand the problem statement, analyze and debug the given code.

**Task 2: Create a Bank Account Management System**

**Objective**: The goal of this assignment is to design a simple Bank Account Management System using the principles of Object-Oriented Programming (OOP). This will help you apply the concepts of classes, objects, inheritance, and encapsulation that you learned in today's lecture.

**Task Details**:

You are tasked to create classes and objects for different types of bank accounts in a bank. Each bank account has common attributes like account number, account holder's name, and balance. There are two types of accounts: Savings Account and Checking Account.

**Instructions**:

1. **BankAccount Class**: Create a base class 'BankAccount' with the following attributes: account number, account holder's name, and balance. This class should have methods to 'Deposit()' and 'Withdraw()' money. Also, include a method 'DisplayAccountInfo()' to display the account's information.
2. **SavingsAccount Class**: Create a subclass 'SavingsAccount' that inherits from the 'BankAccount' class. Add an additional attribute 'interestRate'. Override the 'Deposit()' method to add interest to the account balance whenever a deposit is made.
3. **CheckingAccount Class**: Create another subclass 'CheckingAccount' that also inherits from the 'BankAccount' class. Override the 'Withdraw()' method such that an error message is displayed if the withdrawal amount exceeds the account balance.
4. **Bank Class**: Create a class 'Bank' that has a list of BankAccounts. This class should have methods to 'AddAccount()', 'DepositToAccount()', and 'WithdrawFromAccount()'.
5. **Main Program**: In your main program, create an instance of the 'Bank' class. Create instances of your 'SavingsAccount' and 'CheckingAccount' classes and add them to the bank. Then make some deposits and withdrawals to show how the system works.