--Level -2 Programs---

WAP for the below requirements

a. Create 3 classes.
    a. Player Object
    b. Volleyball Object
    c. MainClass Object
b. Player Object contains static method play()
c. Volley Object contains static method move()
d. MainClass Object contains static method main
e. Main method will play() and move() method.

WAP for the below requirements

1. Create 3 classes
    a. Mobile Object
    b. Charger Object
    c. MainClass Object
2. Mobile Object contains non-static method storePower()
3. Charger Object contains non-static method providePower()
4. MainClass Object contains static method main
5. Main will call storePower() and providePower() method.

WAP for the below requirements

1. Create 3 classes.
    a. Author Object
    b. Book Object
    c. MainClass Object
2. Author can imagine to write story and review page by page
3. Book can hold story and navigate page to page
4. holdStory will call writeStroy
5. navigatePage will call reviewStory
6. main method will call holdStory and navigatePage

Circle and Calculator Program:

a. Create three classes in Java: Circle, Calculator, and MainClass.
b. Define two non-static methods in the Circle class: calculateArea() and calculateCircumference().
c. Implement two static methods in the Calculator class: findSquare() and findCube().
d. findSquare() will call calculateArea()
e. calculateCircumference() will call findCube()
d. In the MainClass, write a static method main.
e. main will findSquare() and calculateCircumference()

Student and Classroom Program:

a. Write a Java program with three classes: Student, Classroom, and MainClass.
b. Include two non-static methods in the Student class: takeExam() and submitAssignment().
c. Implement two static methods in the Classroom class: calculateAverageScore() and conductDiscussion().
d. In the Student class, allow takeExam() to call calculateAverageScore().

e. In the Classroom class, allow calculateAverageScore() to call submitAssignment().
f. In the MainClass, write a static method main.
g. In the main method call takeExam()

WAP for the below requirements

a. Write a Java program with three classes: Library, Book, and MainClass.
b. Include three non-static methods in the Library class: borrowBook(), returnBook(), and checkAvailability().
c. Implement 1 static method and 1 non-static method in the Book class: displayDetails() and reserveBook().
d. In the Library class, allow borrowBook() to call displayDetails().
e. In the Book class, allow displayDetails() to call checkAvailability().
f. In the Book class, allow reserveBook() to call returnBook().
g. In the MainClass, write a static method main.
h. In the main method, call borrowBook() and reserveBook()

WAP for the below requirements

1. Write a Java program with four classes: Store, Product, Category, and MainClass.
2. Include three non-static methods in the Store class: sellProduct(), restockProduct(), and checkInventory().
3. Implement 2 static method in the Product class: displayDetails() and orderProduct().
4. In the Category class, implement 1 static method and 1 non-static method: createCategory() and categorizeProduct().
5. In the Store class, allow sellProduct() to call displayDetails().
6. In the Product class, allow displayDetails() to call checkInventory().
7. In the Product class, allow orderProduct() to call restockProduct().
8. In the Category class, allow createCategory() to call sellProduct().
9. In the Category class, allow categorizeProduct() to call orderProduct().
10. In the MainClass, write a static method main.
11. In the main method, call sellProduct(), orderProduct(), and categorizeProduct().

WAP for the below requirements

1. Write a Java program with four classes: Bank, Account, Transaction, and MainClass.
2. Include 2 non-static methods and 1 static in the Bank class: openAccount(), closeAccount(), and checkBalance().
3. Implement 2 static methods in the Account class: displayAccountDetails() and transferFunds().
4. In the Transaction class, implement 1 static method and 1 non-static method: logTransaction() and processTransaction().
5. In the Bank class, allow openAccount() to call displayAccountDetails().
6. In the Account class, allow displayAccountDetails() to call checkBalance().
7. In the Bank class, allow checkBalance() to call transferFunds().
8. In the Transaction class, allow logTransaction() to call openAccount().
9. In the Transaction class, allow processTransaction() to call closeAccount().
10. In the MainClass, write a static method main.
11. In the main method, call logTransaction and processTransaction().