# **Experiment 5**

1. **Animals (Grouping Related Classes):**

* **Create classes for Dog and Cat.**
* **Put them in a package called pets to group them together because they're both pets.**

# **Program:**

import pets.Dog;

import pets.Cat;

public class Exp5\_1{

public static void main(String args[]){

    Dog d1=new Dog();

    Cat c1=new Cat();

    d1.display();

    d1.bark();

    c1.display();

    c1.sound();

}

}

package pets;

public class Cat{

    public void sound(){

        System.out.println("Meaw Meaw.... Meaw..");

    }

    public void display(){

        System.out.println("I am Cat");

    }

}

package pets;

public class Dog{

    public void bark(){

        System.out.println("Bhau Bhau... Bhau Bhau..");

    }

    public void display(){

        System.out.println("I am Dog");

    }

}

# **Output:**

I am Dog

Bhau Bhau... Bhau Bhau..

I am Cat

Meaw Meaw.... Meaw..

1. **Shape Hierarchy (Avoiding Name Conflicts):**

* **Create two classes named Circle and Rectangle with their respective properties and methods.**
* **Imagine you're working on a separate project that also has a class named Circle for unrelated functionality (e.g., representing a social circle).**
* **Without packages, using both classes in the same program would lead to a conflict.**
* **Solution: Organize your Circle and Rectangle classes within a package named geometry. This enables you to reference them using geometry.Circle and geometry.Rectangle, preventing conflicts with classes from other projects.**

# **Program:**

 import geometry.Circle;

 import geometry.Rectangle;

 public class Exp5\_2 {

     public static void main(String args[]){

         Circle c1 = new Circle(); // Using Circle from the geometry package

         Rectangle r1 = new Rectangle(); // Using Rectangle from the geometry package

         c1.displayArea();

         r1.displayArea();

     }

 }

package geometry;

import java.util.Scanner;

public class Circle {

    public void displayArea() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter radius of circle");

        int radius = sc.nextInt();

        System.out.println("Area of circle is " + (3.14 \* radius \* radius));

    }

}

package geometry;

import java.util.Scanner;

public class Rectangle{

    public void displayArea(){

        Scanner sc= new Scanner(System.in);

        System.out.println("Enter length and breadth of Rectangle");

        int length=sc.nextInt();

        int breadth=sc.nextInt();

        System.out.println("Area of Rectangle is"+(length\*breadth));

    }

}

# **Output:**

Enter radius of circle

7

Area of circle is 153.86

Enter length and breadth of Rectangle

5 6

Area of Rectangle is30

**3. Library Management (Organization and Access Control):**

* **Design classes for Book, Author, and Library.**
* **Create a package named libraryManagement.**
* **Place the Book, Author, and Library classes within this package.**
* **Within the libraryManagement package, define a class named Utils with helper methods like searchBooksByTitle or calculateLateFees.**
* **Mark the Utils class methods as protected to restrict direct access from outside the package. Other classes within libraryManagement can utilize these methods, promoting modularity and controlled access.**

# **Program:**

package libraryManagement;

public class Book {

    String title;

    private Author author;

    private int numberOfPages;

    public Book(String title, Author author) {

        this.title = title;

        this.author = author;

    }

    public Book(String title, Author author, int numberOfPages) {

        this.title = title;

        this.author = author;

        this.numberOfPages = numberOfPages;

    }

    public Book(String title) {

        this.title = title;

    }

    public String getTitle() {

        return title;

    }

    public Author getAuthor() {

        return author;

    }

}

package libraryManagement;

public class Author {

    String author;

    public Author(String author) {

        this.author = author;

    }

    public String getAuthor() {

        return author;

    }

}

package libraryManagement;

import java.util.ArrayList;

import java.util.List;

public class Library {

    private static ArrayList<Book> books = new ArrayList<>();

    private Library() {

    }

    public static void addBook(Book book) {

        books.add(book);

    }

    public static List<Book> getBooks() {

        return new ArrayList<>(books);

    }

    public static Library getInstance() {

        return new Library();  } }

package libraryManagement;

import java.util.\*;

public class Utils {

    public static List<Book> searchBooksByTitle(Library library, String title) {

        List<Book> foundBooks = new ArrayList<>();

        for (Book book : library.getBooks()) {

            if (book.getTitle().equalsIgnoreCase(title)) {

                foundBooks.add(book);

            }

        }

        return foundBooks;

    }

    public static double calculateLateFees(int daysLate) {

        // Logic to calculate late fees based on the number of days late

        return daysLate \* 0.50; // Assuming $0.50 per day late fee

    }

    public static void Showbooks() {

        List<Book> allBooks = Library.getBooks();

        for (Book book : allBooks) {

            System.out.println(book.getTitle() + " by " + book.getAuthor().getAuthor());

        }

    }

}

import libraryManagement.\*;

import java.util.\*;

public class Exp5\_3 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        String bookName;

        Library library = Library.getInstance(); // Get instance of Library

        Book book1 = new Book("The Lord of the Rings", new Author("J.R.R. Tolkien"));

        Book book2 = new Book("Can't Hurt Me", new Author("David Goggins"));

        library.addBook(book1); // Add book1 to library

        library.addBook(book2); // Add book2 to library

        int choice;

        do {

            System.out.print("\nEnter Choice");

            System.out.println("\n1.Add Book\n2.Search Books By Title\n3.Calculate Late Fees\n4.Show All Books\n5.Exit");

            choice = sc.nextInt();

            sc.nextLine(); // Consume newline character

            switch (choice) {

                case 1:

                    System.out.println("Enter Name of Book:");

                    bookName = sc.nextLine();

                    System.out.println("Enter Author Name:");

                    String bookAuthor = sc.nextLine();

                    System.out.println("Enter Page Numbers:");

                    int pageNumbers = sc.nextInt();

                    Author author = new Author(bookAuthor);

                    Book newBook = new Book(bookName, author, pageNumbers);

                    library.addBook(newBook); // Add the new book to the library

                    break;

                case 2:

                    System.out.println("Enter Book name:");

                    bookName = sc.nextLine();

                    List<Book> searchResult = Utils.searchBooksByTitle(library, bookName);

                    for (Book book : searchResult) {

                        System.out.println(book.getTitle() + " by " + book.getAuthor().getAuthor());

                    }

                    break;

                case 3:

                    System.out.println("Enter Late days:");

                    int lateDays = sc.nextInt();

                    double lateFees = Utils.calculateLateFees(lateDays);

                    System.out.println("Late Fees: $" + lateFees);

                    break;

                case 4:

                    Utils.Showbooks();

                    break;

                case 5:

                    System.out.println("Exiting....");

                    break;

            }

        } while (choice != 5);

    }

}

# **Output:**

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

4

The Lord of the Rings by J.R.R. Tolkien

Can't Hurt Me by David Goggins

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

1

Enter Name of Book:

Yugandhar

Enter Author Name:

Shivaji Sawant

Enter Page Numbers:

450

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

4

The Lord of the Rings by J.R.R. Tolkien

Can't Hurt Me by David Goggins

Yugandhar by Shivaji Sawant

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

3

Enter Late days:

4

Late Fees: $2.0

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

2

Enter Book name:

Yugandhar

Yugandhar by Shivaji Sawant

Enter Choice

1.Add Book

2.Search Books By Title

3.Calculate Late Fees

4.Show All Books

5.Exit

5

Exiting....

**4. Online Store (Code Reusability and Maintainability):**

**Develop classes for Product, ShoppingCart, and Order,**

**Create a package named ecommerce.**

**Include the Product, Shopping Cart, and order classes within this package.**

**Consider creating a separate package named paymentProcessing for classes like Payment Gateway and Credit Card if your store handles online payments.**

**This separation allows you to reuse the core functionalities (Product, Shopping Cart, Order) in different parts of your application while keeping payment-related code organized and encapsulated.**

**.**

**Package No 1: ecommerce**

**1.**

package ecommerce;

import java.util.ArrayList;

public class Order {

    private ArrayList<Product> products = new ArrayList<>();

    private String customerName;

    private String shippingAddress;

    private double totalCost;

    private String status;

    public Order(String customerName, String shippingAddress) {

        this.customerName = customerName;

        this.shippingAddress = shippingAddress;

        this.status = "Pending";

    }

    public void addProduct(Product product) {

        products.add(product);

        totalCost += product.getPrice();

    }

    public void cancelOrder() {

        status = "Cancelled";

        System.out.println("Order cancelled successfully.");

    }

    public void updateOrderStatus(String newStatus) {

        status = newStatus;

        System.out.println("Order status updated to: " + newStatus);

    }

}

**2.**

package ecommerce;

import java.util.ArrayList;

public class Product {

    private String name;

    private int price;

    private static ArrayList<Product> productList = new ArrayList<>();

    public Product(String name, int price) {

        this.name = name;

        this.price = price;

        productList.add(this); // Add the new product to the productList

    }

    public String getName() {

        return name;

    }

    public int getPrice() {

        return price;

    }

    public static void showProducts() {

        for (Product product : productList) {

          System.out.println(product.getName() + " price " + product.getPrice());

        }

      }

    public void updateProductDetails(int newPrice) {

        this.price = newPrice;

        System.out.println("Product details updated successfully.");

    }

}

**3.**

package ecommerce;

import java.util.ArrayList;

public class ShoppingCart

 {

    private ArrayList<Product> products = new ArrayList<>();

    public void showListOfProducts() {

        System.out.println("Name\tPrice");

        for (Product product : products) {

            System.out.println(product.getName() + "\t" + product.getPrice());

        }

    }

    public void addProduct(Product product) {

        products.add(product);

        System.out.println("Product added to cart: " + product.getName());

    }

    public void removeProduct(Product product) {

        products.remove(product);

        System.out.println("Product removed from cart: " + product.getName());

    }

    public double calculateTotalCost() {

        double totalCost = 0;

        for (Product product : products) {

            totalCost += product.getPrice();

        }

        return totalCost;

    }

}

**Package No2: paymentProcessing**

**1.**

package paymentProcessing;

public class CreditCard {

    private String cardNumber;

    private String expirationDate;

    private String cardholderName;

    private int cvv;

    public CreditCard(String cardNumber, String expirationDate, String cardholderName, int cvv) {

        this.cardNumber = cardNumber;

        this.expirationDate = expirationDate;

        this.cardholderName = cardholderName;

        this.cvv = cvv;

    }

    public boolean validateCardNumber(String cardNumber) {

        // Check if card number is not null and has 16 digits

        if (cardNumber == null || cardNumber.length() != 16) {

            return false;

        }

        // Convert card number to array of digits

        char[] digits = cardNumber.toCharArray();

        // Use Luhn algorithm to validate the credit card number

        int sum = 0;

        for (int i = digits.length - 1; i >= 0; i--) {

            int digit = Character.getNumericValue(digits[i]);

            if ((digits.length - i) % 2 == 0) {

                digit \*= 2;

                if (digit > 9) {

                    digit -= 9;

                }

            }

            sum += digit;

        }

        return sum % 10 == 0;

    }

    public boolean authorizePayment(double amount) {

        // Basic authorization: if the amount is less than 1000, authorize the payment

        return amount <= 1000;

    }

    public boolean processRefund(double amount) {

        // Basic refund processing: if the amount is positive, process the refund

        return amount > 0;

    }

}

**2.**

package paymentProcessing;

public class PaymentGateway {

    public void processPayment(double amount, double totalCost) {

        // Implementation of payment processing logic

        if (amount == totalCost) {

            System.out.println("Paid Successfully");

        }

        else{

            System.out.println("Enter valid amount");

        }

    }

    public void handlePaymentResponse(boolean isSuccess) {

        // Implementation to handle payment response

        if (isSuccess) {

            System.out.println("Payment Received");

        }

    }

}

**Main function**

import ecommerce.\*;

import paymentProcessing.\*;

import java.util.Scanner;

public class Exp5\_4 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Product p1 = new Product("Sugar", 40);

        Product p2 = new Product("Rice", 70);

        Product p3 = new Product("Wheat", 42);

        Product p4 = new Product("Potato", 40);

        ShoppingCart s1 = new ShoppingCart();

        PaymentGateway paymentGateway = new PaymentGateway();

        double totalCost = 0;

        int choice;

        do {

            System.out.println("\nEnter Choice:");

            System.out.println("1. Show all products\n2. Add products to cart\n3. Calculate Total Cost\n4. Pay Amount\n5. Exit");

            choice = sc.nextInt();

            switch (choice) {

                case 1:

                    Product.showProducts();

                    break;

                case 2:

                    System.out.println("Enter Serial Number to add product");

                    int serialNo = sc.nextInt();

                    switch (serialNo) {

                        case 1:

                            s1.addProduct(p1);

                            break;

                        case 2:

                            s1.addProduct(p2);

                            break;

                        case 3:

                            s1.addProduct(p3);

                            break;

                        case 4:

                            s1.addProduct(p4);

                            break;

                        default:

                            System.out.println("Invalid Serial Number.");

                    }

                    break;

                case 3:

                    totalCost = s1.calculateTotalCost();

                    System.out.println("Total cost: " + totalCost);

                    break;

                    case 4:

                    System.out.println("Enter Amount to pay");

                    int amountPaying = sc.nextInt();

                    sc.nextLine(); // Consume the newline character

                    System.out.println("Enter Credit Card Number (16 digit)");

                    String cardNumber = sc.nextLine();

                    System.out.print("Enter Expiry Date:");

                    String expirationDate = sc.nextLine();

                    System.out.println("Enter Card Holder Name:");

                    String cardholderName = sc.nextLine();

                    System.out.println("Enter CVV:");

                    int cvv = sc.nextInt();

                    CreditCard creditCard = new CreditCard(cardNumber, expirationDate, cardholderName, cvv);

                    // Call validateCardDetails() method to check if the card number is valid

                     boolean isCardValid = creditCard.validateCardNumber(cardNumber);

                     if (!isCardValid) {

                         System.out.println("Invalid credit card number format. Please enter a 16-digit card number.");

                         break; // Exit the switch statement

                     }

                    paymentGateway.processPayment(amountPaying, totalCost);

                    break;

                case 5:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice. Please try again.");

            }

        } while (choice != 5);

        sc.close(); // Close the scanner

    }

}

# **Output:**

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

1

Sugar price 40

Rice price 70

Wheat price 42

Potato price 40

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

2

Enter Serial Number to add product

1

Product added to cart: Sugar

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

2

Enter Serial Number to add product

2

Product added to cart: Rice

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

3

Total cost: 110.0

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

4

Enter Amount to pay

110

Enter Credit Card Number (16 digit)

123456780123456

Enter Expiry Date:12/06/2030

Enter Card Holder Name:

Prathmesh Patil

Enter CVV:

4567

Invalid credit card number format. Please enter a 16-digit card number.

Enter Choice:

1. Show all products

2. Add products to cart

3. Calculate Total Cost

4. Pay Amount

5. Exit

5

Exiting...

PS D:\pull from github\Java\Experiment\_5>