Lab Examination Question Paper

ACTS TRAINING CENTRE – Bangalore Question Code : **ABC**

DBDA AUG 2024

Core Java

Date : 15.10.2024 Duration : 2 Hrs Max. Marks : 40

3) Problem Statement: Product Management System (Using JDBC in Java)
Design and implement a Product Management System using Java and JDBC to interact with a relational database. The system should allow users to manage products in an inventory, including operations such as adding new products, updating product details, viewing all products, and searching for a product by its name or category. Additionally, implement a business logic to calculate the total value of the inventory based on product prices and quantities.

Requirements:
Database Table:
Create a table called Products in your database with the following columns:
productId (Primary Key, int)
productName (String)
category (String)
price (double)
quantity (int)
Operations to be Performed:

Add a New Product:
Implement a method addProduct() to insert new products to the Products table in the database.
Update Product Details:
Implement a method updateProduct(Product product) to update existing product information (e.g., price, quantity, or category) in the database.
View All Products:
Implement a method viewAllProducts() that fetches and displays all products from the Products table.
Search for a Product by Name or Category:
Implement a method searchProduct(String keyword) that searches for products either by name or category (use the keyword to search both fields) and displays the matching products.
Business Logic:

Calculate Total Inventory Value:
Implement a method calculateTotalInventoryValue() that calculates and returns the total value of the inventory by multiplying each product's price by its available quantity, summing up the result for all products.
Constraints:
Use JDBC to connect to the database, and perform CRUD operations (Create, Read, Update, Delete) on the Products table.

Lab Examination Question Paper

ACTS TRAINING CENTRE – Bangalore Question Code : **DBC**

DBDA AUG 2024

Core Java

Date : 15.10.2024 Duration : 2 Hrs Max. Marks : 40

Problem Statement: Product Management System (Using OOP, Hierarchical Inheritance, and Interface in Java)
Design and implement a Product Management System using Object-Oriented Programming principles, hierarchical inheritance, and an interface in Java. The system should allow for the management of different types of products, such as electronics, groceries, and furniture, using inheritance. Additionally, there should be an interface to handle product discount calculations, and a business logic that calculates the total value of products in the inventory.

Requirements:
Product Class (Base Class):

Attributes:
int productId: A unique identifier for each product.
String productName: The name of the product.
double price: The price of the product.
int quantity: The quantity available in the inventory.
Methods:
Constructor to initialize product attributes.
Getter and setter methods for all attributes.
displayProductDetails(): Method to display the details of a product.

Electronics Class (Derived Class):
Inherits from the Product class.
Additional Attributes:
String warrantyPeriod: Warranty period of the electronics product.
Methods:
Constructor to initialize attributes.
Override displayProductDetails() to include the warranty period.

Groceries Class (Derived Class):
Inherits from the Product class.
Additional Attributes:
String expiryDate: Expiry date of the grocery item.
Methods:

Constructor to initialize attributes.
Override displayProductDetails() to include the expiry date.

Furniture Class (Derived Class):
Inherits from the Product class.
Additional Attributes:
String material: Material used for the furniture (e.g., wood, metal).
Methods:
Constructor to initialize attributes.
Override displayProductDetails() to include the material.
Interface: Discountable:

Method double calculateDiscount(): This method should be implemented by all product types and should return a discount based on the type of product.
Electronics: 10% discount.
Groceries: 5% discount.
Furniture: 15% discount.
Operations to be Performed:

Create object of Electronics, Groceries, or Furniture product
displays the details of all products in the inventory.
calculate the discount of all product
Calculate Total Inventory Value with Discount:
Implement a method calculateTotalInventoryValueWithDiscount() that calculates the total value of the inventory, taking into account the discounts for each product. The formula for each product is (price - discount) * quantity.

Lab Examination Question Paper

ACTS TRAINING CENTRE – Bangalore Question Code : **EBC**

DBDA AUG 2024

Core Java

Date : 15.10.2024 Duration : 2 Hrs Max. Marks : 40

Problem Statement: Product Management System Using File Handling (Object-Based Read/Write in Java)

Design and implement a **Product Management System** using Java where product data is stored and retrieved from a file using **object-based read and write operations**. The system should allow the user to perform operations such as adding new products, viewing all products, and performing business logic (calculating the total inventory value of all products).

Requirements:

1. **Product Class:**
    - Attributes:
        - `int productId`: A unique identifier for each product.
        - `String productName`: Name of the product.
        - `double price`: The price of the product.
        - `int quantity`: The quantity of the product in stock.
    - Methods:
        - Constructor to initialize product details.
        - `displayProductDetails()`: Method to display product details.
        - Getter and setter methods for all attributes.
2. **File Handling:**
    - **Storing Products in a File:**
      Products should be serialized and saved in a file called `products.dat`. Each product object should be written to the file using object serialization.
    - **Reading Products from a File:**
      The system should be able to deserialize and load the list of products from `products.dat`.
3. **Operations to be Performed:**
    - **Add New Product:**
      Implement a method `addProduct(Product product)` that allows

adding a new product to the list and saves it to the file.

- **View All Products:**
  Implement a method `viewAllProducts()` that reads the list of products from the file and displays each product's details.
- **Business Logic - Calculate Total Inventory Value:**
  Implement a method `calculateTotalInventoryValue()` to calculate and return the total value of all products in inventory. The value is computed as `price * quantity` for each product.

Lab Examination Question Paper

ACTS TRAINING CENTRE – Bangalore Question Code : **FBC**

DBDA AUG 2024

Core Java

Date : 15.10.2024 Duration : 2 Hrs Max. Marks : 40

Problem Statement: Product Management System Using HashMap (Object-Based with Business Logic in Java)

You are tasked with developing a **Product Management System** using **HashMap** in Java where each product is stored as an object and mapped by its unique product ID. The system should allow users to manage a collection of products, perform CRUD operations, and implement business logic beyond basic CRUD. Specifically, the system should allow sorting products by price and calculating the total value of all products in inventory.

Requirements:

1. **Product Class:**
    - Attributes:
        - `int productId`: A unique identifier for each product.
        - `String productName`: Name of the product.
        - `double price`: The price of the product.
        - `int quantity`: Quantity of the product in stock.
    - Methods:
        - Constructor to initialize product details.
        - `displayProductDetails()`: Method to display product details.
        - Getter and setter methods for all attributes.
2. **HashMap to Store Products:**
    - Use a **HashMap** where:
        - The key is `productId` (int).
        - The value is a `Product` object.
    - This structure allows efficient retrieval, update, and deletion of products based on their unique ID.
3. **Operations to be Performed:**
    - **Add New Product:**
      Implement a method `addProduct(Product product)` that adds a

new product to the HashMap.

- **View All Products:**
  Implement a method `viewAllProducts()` to display the details of all products.
- **Update Product:**
  Implement a method `updateProduct(int productId)` to update the details of a specific product.
- **Delete Product:**
  Implement a method `deleteProduct(int productId)` to remove a product from the HashMap by its ID.

4. **Business Logic Operations:**

- **Calculate Total Inventory Value:**
  Implement a method `calculateTotalInventoryValue()` that calculates the total value of all products in inventory, defined as `price * quantity` for each product.