CSA0961- JAVA

TEST – 7

SET 2

1. Design a class BankAccount with properties accountNumber and balance, and methods deposit() and withdraw(). Extend this class with subclasses SavingsAccount and CheckingAccount. Implement specific rules such as minimum balance requirements and interest calculation for savings accounts.

PROGRAM :

```
class BankAccount {
    protected String accountNumber;
    protected double balance;

    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.printf("Deposited $%.2f. New balance is $%.2f.%n", amount, balance);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0) {
            if (amount <= balance) {
                balance -= amount;
```

```
            System.out.printf("Withdrew $%.2f. New balance is $%.2f.%n", amount, balance);
        } else {
            System.out.println("Insufficient funds.");
        }
    } else {
        System.out.println("Withdrawal amount must be positive.");
    }
}

public double getBalance() {
    return balance;
}
}

class SavingsAccount extends BankAccount {
private static final double MIN_BALANCE = 100;
private static final double INTEREST_RATE = 0.02;

public SavingsAccount(String accountNumber, double balance) {
    super(accountNumber, balance);
}

@Override
public void withdraw(double amount) {
    if (balance - amount >= MIN_BALANCE) {
        super.withdraw(amount);
    } else {
        System.out.printf("Cannot withdraw $%.2f. Minimum balance requirement of $%.2f not
met.%n", amount, MIN_BALANCE);
    }
}

public void applyInterest() {
```

```
        double interest = balance * INTEREST_RATE;
        deposit(interest);
        System.out.printf("Applied interest of $%.2f. New balance is $%.2f.%n", interest, balance);
    }
}

class CheckingAccount extends BankAccount {
    private static final double OVERDRAFT_LIMIT = -500;

    public CheckingAccount(String accountNumber, double balance) {
        super(accountNumber, balance);
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= OVERDRAFT_LIMIT) {
            super.withdraw(amount);
        } else {
            System.out.printf("Cannot withdraw $%.2f. Overdraft limit of $%.2f exceeded.%n", amount,
OVERDRAFT_LIMIT);
        }
    }
}
public class BankAccountDemo {

    public static void main(String[] args) {

        SavingsAccount savings = new SavingsAccount("SA123", 1500);
        CheckingAccount checking = new CheckingAccount("CA123", 200);

        savings.deposit(500);
        savings.withdraw(200);
        savings.applyInterest();
```

```
        checking.deposit(100);
        checking.withdraw(250);
        checking.withdraw(600);
    }
}
```

OUTPUT :

```
Deposited $500.00. New balance is $2000.00.
Withdrew $200.00. New balance is $1800.00.
Deposited $36.00. New balance is $1836.00.
Applied interest of $36.00. New balance is $1836.00.
Deposited $100.00. New balance is $300.00.
Withdrew $250.00. New balance is $50.00.
Cannot withdraw $600.00. Overdraft limit of $-500.00 exceeded.
```

3. Design a class Product with properties productId, name, and price. Extend this class with subclasses Electronics and Clothing. Implement methods to calculate discounts based on membership status for electronics and seasonal sales for clothing.

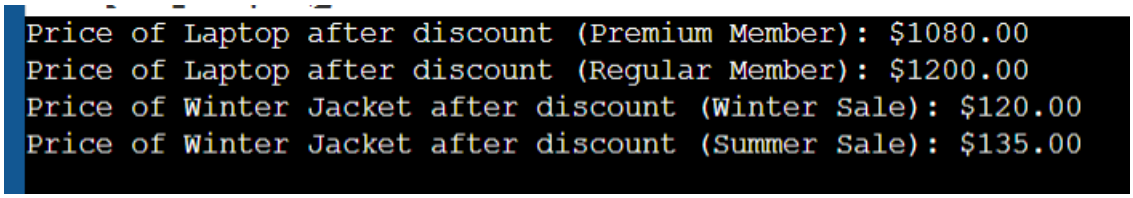
PROGRAM :

```
class Product {
    protected String productId;
    protected String name;
    protected double price;

    public Product(String productId, String name, double price) {
        this.productId = productId;
        this.name = name;
        this.price = price;
    }

    public String getProductId() {
        return productId;
    }
```

```
    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }
}

class Electronics extends Product {

    private static final double MEMBER_DISCOUNT = 0.10; // 10% discount for members

    public Electronics(String productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(boolean isPremiumMember) {
        double discount = isPremiumMember ? MEMBER_DISCOUNT : 0;
        return price * (1 - discount);
    }
}

class Clothing extends Product {

    private static final double WINTER_SALE_DISCOUNT = 0.20;
    private static final double SUMMER_SALE_DISCOUNT = 0.10;

    public Clothing(String productId, String name, double price) {
        super(productId, name, price);
    }
```

```
    public double applySeasonalSale(String season) {
        double discount;
        switch (season) {
            case "Winter":
                discount = WINTER_SALE_DISCOUNT;
                break;
            case "Summer":
                discount = SUMMER_SALE_DISCOUNT;
                break;
            default:
                discount = 0;
                break;
        }
        return price * (1 - discount);
    }
}
public class ProductDemo {

    public static void main(String[] args) {

        Electronics laptop = new Electronics("E123", "Laptop", 1200.00);
        Clothing jacket = new Clothing("C456", "Winter Jacket", 150.00);

        System.out.printf("Price of %s after discount (Premium Member): $%.2f%n", laptop.getName(),
laptop.calculateDiscount(true));
        System.out.printf("Price of %s after discount (Regular Member): $%.2f%n", laptop.getName(),
laptop.calculateDiscount(false));

        System.out.printf("Price of %s after discount (Winter Sale): $%.2f%n", jacket.getName(),
jacket.applySeasonalSale("Winter"));
        System.out.printf("Price of %s after discount (Summer Sale): $%.2f%n", jacket.getName(),
jacket.applySeasonalSale("Summer"));
    }
}
```

OUTPUT :



4. Design a class LibraryItem with properties title, author, and year. Extend this class with subclasses Book and DVD. Implement methods for checking in and out items, and display detailed information for each item type.

PROGRAM :

```
abstract class LibraryItem {
    protected String title;
    protected String author;
    protected int year;
    protected boolean isCheckedOut;

    public LibraryItem(String title, String author, int year) {
        this.title = title;
        this.author = author;
        this.year = year;
        this.isCheckedOut = false;
    }

    public abstract void displayInfo();

    public void checkOut() {
        if (isCheckedOut) {
            System.out.println("Item is already checked out.");
        } else {
            isCheckedOut = true;
            System.out.println("Item checked out: " + title);
        }
    }
```

```
    public void checkIn() {
        if (!isCheckedOut) {
            System.out.println("Item is not checked out.");
        } else {
            isCheckedOut = false;
            System.out.println("Item checked in: " + title);
        }
    }
}

class Book extends LibraryItem {
    private String genre;

    public Book(String title, String author, int year, String genre) {
        super(title, author, year);
        this.genre = genre;
    }

    @Override
    public void displayInfo() {
        System.out.printf("Book Title: %s%nAuthor: %s%nYear: %d%nGenre: %s%nChecked Out:
%b%n",
                title, author, year, genre, isCheckedOut);
    }
}

class DVD extends LibraryItem {
    private String genre;

    public DVD(String title, String author, int year, String genre) {
        super(title, author, year);
        this.genre = genre;
```

```
    }

    @Override
    public void displayInfo() {
        System.out.printf("DVD Title: %s%nDirector: %s%nYear: %d%nGenre: %s%nChecked Out:
%b%n",
                title, author, year, genre, isCheckedOut);
    }
}
public class LibraryDemo {

    public static void main(String[] args) {

        Book book = new Book("1984", "George Orwell", 1949, "Dystopian");
        DVD dvd = new DVD("Inception", "Christopher Nolan", 2010, "Science Fiction");



        book.checkOut();
        book.displayInfo();
        book.checkIn();

        dvd.checkOut();
        dvd.displayInfo();
        dvd.checkIn();
    }
}
```

OUTPUT :

```
Item checked out: 1984
Book Title: 1984
Author: George Orwell
Year: 1949
Genre: Dystopian
Checked Out: true
Item checked in: 1984
Item checked out: Inception
DVD Title: Inception
Director: Christopher Nolan
Year: 2010
Genre: Science Fiction
Checked Out: true
Item checked in: Inception
```