UNIVERSITI
TEKNOLOGI
PETRONAS

Course Code
Object and Oriented Programming
Semester May 2024
Group Number

Lecturer:

Name	Matric ID	Course
Lydia Natasha Binti Muharral	24000481	Information Technology
Wardina Saffiya Binti Jamalulil	24000996	Information Technology
Wan Nur Irdina bt Wan Hasbullah	24000247	Information Technology
Nur Aisya' Sofea Binti Husin	22012332	Information Technology
Muhammad Aiman Haikal bin Mohammad Akmal Surish	24000458	Computer Science

Submission Date: 2 June 2024

Total Marks:

QUESTION

OOP Homework 1
Due Date: 2 June 2024, 11.59PM
ATM
Write dummy classes (blank methods) for the following:

1) Client:
    data: name, userid, password, phone number, address etc
2) Transaction:
    data: amount, balance, date
3) Account:
    data: client, acc number, current balance, transaction list (history)
4) Clients:
    data: list of clients
    actions: add, remove, print, find
5) Accounts:
    data: list of accounts
    actions: add, remove, print, find
Write code to test the class

ANSWER:

No.	Answer (Java Language)

1.	public class Client {
	  private String name;
	  private String userid;
	  private String password;
	  private String phoneNumber;
	  private String address;
	
	  // Constructor
	  public Client(String name, String userid, String password, String phoneNumber, String address) {
	  this.name = name;
	  this.userid = userid;
	  this.password = password;
	  this.phoneNumber = phoneNumber;
	  this.address = address;
	  }
	
	  // Getters and setters for each field
	  public String getName() { return name; }
	  public void setName(String name) { this.name = name; }

	public String getUserid() { return userid; }
	public void setUserid(String userid) { this.userid = userid; }
	
	public String getPassword() { return password; }
	public void setPassword(String password) { this.password = password; }
	
	public String getPhoneNumber() { return phoneNumber; }
	public void setPhoneNumber(String phoneNumber) { this.phoneNumber = phoneNumber; }
	
	public String getAddress() { return address; }
	public void setAddress(String address) { this.address = address; }
	
	@Override
	public String toString() {
	return "Client{name='" + name + "', userid='" + userid + "', phoneNumber='" + phoneNumber + "', address='" + address + "'}";
	}
	}

2.	class Transaction {
	private double amount;
	private double balance;
	private String date;
	
	public Transaction(double amount, double balance, String date) {
	
	}
	
	public double getAmount() {
	
	return amount;
	
	}
	public void setAmount(double amount) {
	
	this.amount = amount;
	
	}
	
	public double getBalance() {
	
	return balance;

	}
	public void setBalance(double balance) {
	
	this.balance = balance; }
	
	public String getDate() {
	
	return date;
	
	}
	public void setDate(String date) {
	
	this.date = date;
	
	}
	}

3.	class Account {
	private Client client;
	private String accountNumber;
	private double currentBalance;
	private List<Transaction> transactionList;
	
	// Dummy methods
	public Account(Client client, String accountNumber, double currentBalance) {
	// Initialize fields
	this.transactionList = new ArrayList<>();
	}
	
	// Getters and setters
	public Client getClient() { return client; }
	public void setClient(Client client) { this.client = client; }
	
	public String getAccountNumber() { return accountNumber; }
	public void setAccountNumber(String accountNumber) { this.accountNumber = accountNumber; }
	
	public double getCurrentBalance() { return currentBalance; }
	public void setCurrentBalance(double currentBalance) { this.currentBalance = currentBalance; }
	
	public List<Transaction> getTransactionList() { return transactionList; }
	public void setTransactionList(List<Transaction> transactionList) { this.transactionList = transactionList; }
	
	// Method to add transaction to the transaction list

	public void addTransaction(Transaction transaction) {
	// Add transaction to list
	}
	}

4.	import java.util.ArrayList;
	import java.util.List;
	
	public class Clients {
	private List<Client> clientList;
	
	// Constructor to initialize the client list
	public Clients() {
	clientList = new ArrayList<>();
	}
	
	// Method to add a client to the list
	public void add(Client client) {
	// Add client
	}
	
	// Method to remove a client from the list
	public void remove(String userid) {
	
	}
	
	// Method to print all clients
	public void print() {
	
	}
	
	// Method to find a client in the list
	public Client find(String userid) {
	
	return null;
	}
	}

5.	import java.util.ArrayList;
	import java.util.List;
	import java.util.Optional;
	
	class Account {
	private int id;
	private String name;
	private double balance;
	
	public Account(int id, String name, double balance) {
	this.id = id;

```
        this.name = name;
        this.balance = balance;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public double getBalance() {
        return balance;
    }

    @Override
    public String toString() {
        return "Account{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", balance=" + balance +
                '}';
    }
}

class AccountManager {
    private List<Account> accounts;

    public AccountManager() {
        accounts = new ArrayList<>();
    }

    public void addAccount(Account account) {
        accounts.add(account);
    }

    public void removeAccount(int id) {
        accounts.removeIf(account -> account.getId() == id);
    }

    public void printAllAccounts() {
        accounts.forEach(System.out::println);
    }

    public Account findAccountById(int id) {
        Optional<Account> account = accounts.stream()
                .filter(a -> a.getId() == id)
                .findFirst();
        return account.orElse(null);
```

```
    }
}
```