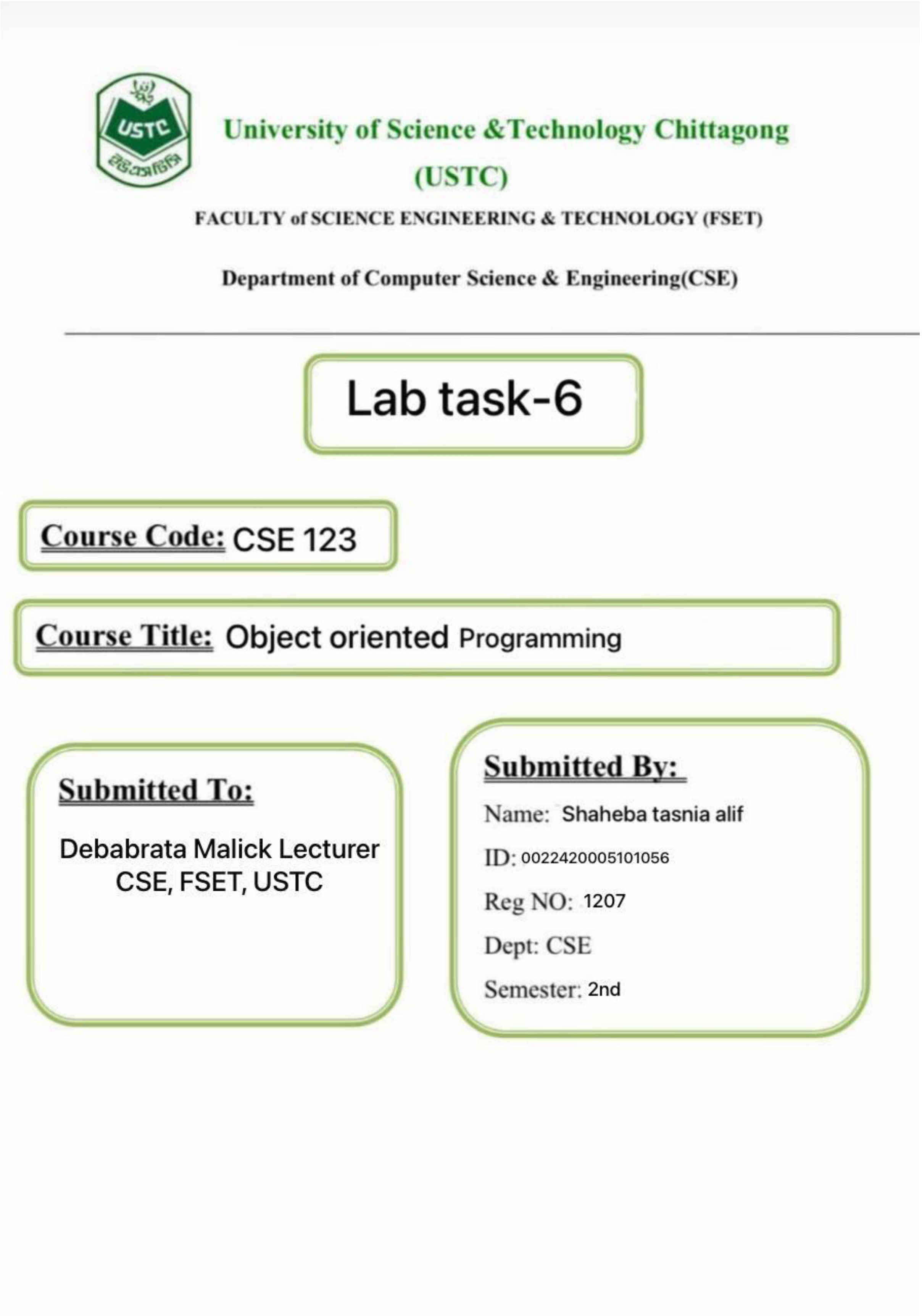

University of Science &Technology Chittagong (USTC)

FACULTY of SCIENCE ENGINEERING & TECHNOLOGY (FSET)

Department of Computer Science & Engineering(CSE)

Lab task-6

Course Code: CSE 123

Course Title: Object oriented Programming

Submitted To:

Debabrata Malick Lecturer
CSE, FSET, USTC

Submitted By:

Name: Shaheba tasnia alif

ID: 0022420005101056

Reg NO: 1207

Dept: CSE

Semester: 2nd

1. Project Title

SmartBudget: A Personal Finance Manager

2. **Project Purpose & Problem Statement**

Purpose:

SmartBudget is a desktop application that helps individuals manage their personal finances by tracking income, expenses, savings, and budgeting goals.

Problem it Solves:

Many people struggle with managing their finances due to a lack of organization or real-time insights into their spending habits. This project provides a user-friendly way to track transactions and analyze spending patterns.

Importance of OOP Concepts:

OOP allows the application to be modular, extensible, and maintainable. It makes it easier to manage financial entities such as accounts, transactions, and budgets.

3. **Project Goals and Key Functionalities**

Goals:

- Help users monitor their income and expenses
- Allow users to set and track budget goals
- Provide simple reports and insights

Key Functionalities:

- Add/Edit/Delete income and expenses
- Categorize transactions (e.g., Food, Rent, Travel)
- View monthly spending summary
- Set monthly budget goals
- Simple console-based UI (upgradeable to GUI later)

4. **Technologies Used**

- Programming Language: Java
- Frameworks/Libraries: None (Core Java)
- Database: No external DB; uses file-based storage (can be upgraded to SQLite)

5. **Use of OOP Principles**

- Encapsulation:

Private fields in classes with public getters and setters to protect data.

- Inheritance:

Transaction superclass with Income and Expense subclasses.

- Polymorphism:

Overriding toString() for better object representation; overloading methods for adding different types of transactions.

- Abstraction:

Interface Storable for saving/loading data to/from files, hiding file I/O details.

6. **Project Phases & Timeline**

Phase	Description	Est. Duration
Phase 1: Planning	Requirement gathering & class design.	1 day
Phase 2: Core Development	Implementing classes & logic	3 days
Phase 3: File I/O	Add saving/loading functionality	1 day
Phase 4: Testing	Test all functionalities.	1 day
Phase 5: Documentation	Proposal & final report	1 day

7. **Final Product Outcomes**

- Track income and expenses by category
- View spending summaries and progress toward budget goals
- Serve as a foundational project that can be expanded to GUI or web app

Usefulness:

This helps users become financially aware, make better spending decisions, and establish savings habits.

8. **Proposal Summary**

SmartBudget is a personal finance manager built with OOP principles to provide a modular and scalable financial tracking tool. It simplifies money management and lays a strong foundation for future upgrades like database integration and UI enhancements.

9. **References**

- Oracle Java Documentation: https://docs.oracle.com/javase/8/docs/
- W3Schools Java OOP: https://www.w3schools.com/java/java_oop.asp

Java Code (Core Structure):

```
abstract class Transaction {
    protected String date;
    protected String category;
    protected double amount;

    public Transaction(String date, String category, double amount) {
        this.date = date;
        this.category = category;
        this.amount = amount;
    }

    public abstract String getType();

    public String toString() {
        return "[" + getType() + "] " + category + ": $" + amount + " on " + date;
    }
```

```
}

class Income extends Transaction {
    public Income(String date, String category, double amount) {
        super(date, category, amount);
    }

    @Override
    public String getType() {
        return "Income";
    }
}

class Expense extends Transaction {
    public Expense(String date, String category, double amount) {
        super(date, category, amount);
    }

    @Override
    public String getType() {
        return "Expense";
    }
}

import java.util.*;

class FinanceManager {
    private List<Transaction> transactions = new ArrayList<>();
    public void addTransaction(Transaction t) {
        transactions.add(t);
    }

    public void viewSummary() {
        double totalIncome = 0, totalExpense = 0;
        for (Transaction t : transactions) {
            if (t instanceof Income) totalIncome += t.amount;
            else totalExpense += t.amount;
        }
        System.out.println("Total Income: $" + totalIncome);
        System.out.println("Total Expense: $" + totalExpense);
        System.out.println("Balance: $" + (totalIncome - totalExpense));
    }

    public void showTransactions() {
        for (Transaction t : transactions) {
            System.out.println(t);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        FinanceManager fm = new FinanceManager();
        fm.addTransaction(new Income("2025-04-01", "Salary", 3000));
        fm.addTransaction(new Expense("2025-04-03", "Groceries", 150));
        fm.addTransaction(new Expense("2025-04-04", "Transport", 50));

        fm.showTransactions();
        fm.viewSummary();
    }
}
```