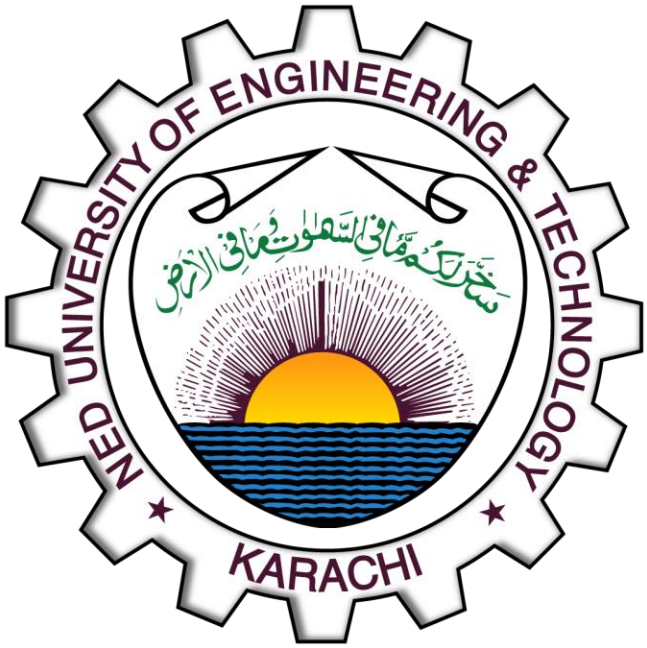
OBJECT ORIENTED PROGRAMMING

[CT-260]



NAME: TAHA AHMED MALLICK

ROLL NO: CT-25183

DEPARTMENT: BCIT **BATCH:** 2025

YEAR & SECTION: FSCS-D

LAB : 1

QUESTION#1:

CODE:

```
#include <iostream>
using namespace std;

class Student
{
public:
    string name;
    int roll_no, sem;
    char sec;
};

int main()
{
    Student students[4];
    cout << "Enter Student Records:\n";
    for (int i = 0; i < 4; i++)
    {
        cout << "Student " << (i + 1) << ":\n";
        cout << "Name: ";
        cin >> students[i].name;
        cout << "Roll No: ";
        cin >> students[i].roll_no;
        cout << "Semester: ";
        cin >> students[i].sem;
        cout << "Section: ";
        cin >> students[i].sec;
    }

    cout << "\n====Students of Section A====\n";
    for (int i = 0; i < 4; i++)
        if (students[i].sec == 'A' || students[i].sec == 'a')
            cout << "Name: " << students[i].name << ", Roll No: " <<
students[i].roll_no << ", Semester: " << students[i].sem << endl;

    return 0;
}
```

OUTPUT:

```
Enter Student Records:
Student 1:
Name: Taha
Roll No: 183
Semester: 2
Section: D
Student 2:
Name: Arslan
Roll No: 184
Semester: 3
Section: A
Student 3:
Name: Aryan
Roll No: 176
Semester: 2
Section: D
Student 4:
Name: Sami
Roll No: 185
Semester: 3
Section: A

====Students of Section A====
Name: Arslan, Roll No: 184, Semester: 3
Name: Sami, Roll No: 185, Semester: 3
```

QUESTION#2:

CODE:

```
#include <iostream>
using namespace std;

class BankAccount
{
private:
    int balance, transactionCount;

public:
    BankAccount(int amt) : balance(amt), transactionCount(0) {}

    int getBalance()
    {
        return balance;
    }
    int getTransactionCount()
    {
        return transactionCount;
    }
    bool deposit(int amt)
    {
        if (amt > 0)
        {
            balance += amt;
            transactionCount++;
            return false;
        }
        return true;
    }
    bool withdraw(int amt)
    {
        if (amt > 0 && amt <= balance)
        {
            balance -= amt;
            transactionCount++;
            return false;
        }
```

```
        return true;
    }
};

int main()
{
    cout << "Welcome to ABC Bank!" << endl;
    BankAccount myAccount(5000);
    int choice;
    do
    {
        cout << "\n\nSelect options from below:" << endl;
        cout << "1. Display balance" << endl;
        cout << "2. Display number of transactions" << endl;
        cout << "3. Display interest earned for this period" << endl;
        cout << "4. Make a deposit" << endl;
        cout << "5. Make a withdrawal" << endl;
        cout << "6. Exit" << endl;
        cin >> choice;
        cout << endl;
        switch (choice)
        {
        case 1:
            cout << "Current balance: ";
            cout << myAccount.getBalance() << endl;
            break;
        case 2:
            cout << "Number of transactions: ";
            cout << myAccount.getTransactionCount() << endl;
            break;
        case 3:
            cout << "Enter annual interest rate (in %): ";
            float rate;
            cin >> rate;
            cout << "Interest earned for this period: ";
            cout << (myAccount.getBalance() * rate / 100) / 12 << endl;
            break;
        case 4:
            cout << "Current balance: ";
            cout << myAccount.getBalance() << endl;
            cout << "Enter amount to deposit: ";
```

```
            int depositAmount;
            cin >> depositAmount;
            if (myAccount.deposit(depositAmount))
                cout << "Invalid deposit amount!" << endl;
            else
            {
                cout << "New balance: ";
                cout << myAccount.getBalance() << endl;
            }
            break;
        case 5:
            cout << "Current balance: ";
            cout << myAccount.getBalance() << endl;
            cout << "Enter amount to withdraw: ";
            int withdrawAmount;
            cin >> withdrawAmount;
            if (myAccount.withdraw(withdrawAmount))
                cout << "Invalid withdrawal amount!" << endl;
            else
            {
                cout << "New balance: ";
                cout << myAccount.getBalance() << endl;
            }
            break;
        case 6:
            cout << "Exiting..." << endl;
            break;
        default:
            cout << "Invalid option" << endl;
            break;
        }
    } while (choice != 6);
    return 0;
}
```

OUTPUT:

```
Welcome to ABC Bank!


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
-0

Invalid option


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
1

Current balance: 5000


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
2

Number of transactions: 0


Select options from below:
```

```
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
3

Enter annual interest rate (in %): 7
Interest earned for this period: 29.1667


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
4

Current balance: 5000
Enter amount to deposit: 65000
New balance: 70000


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
5

Current balance: 70000
Enter amount to withdraw: 10000
New balance: 60000


Select options from below:
```

```
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
2

Number of transactions: 2


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
4

Current balance: 60000
Enter amount to deposit: -500
Invalid deposit amount!


Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
5. Make a withdrawal
6. Exit
2



Select options from below:
1. Display balance
2. Display number of transactions
3. Display interest earned for this period
4. Make a deposit
```

```
5. Make a withdrawal
6. Exit
6

Exiting...
```

QUESTION#3:

CODE:

```
#include <iostream>
#include <cstring>
using namespace std;

class Employee
{
private:
    char *firstName;
    string lastName;
    int salary;

public:
    Employee() : firstName(nullptr), lastName(""), salary(0) {}

    void setInfo(const char *first, string last, int amt)
    {
        firstName = new char[strlen(first) + 1];
        strcpy(firstName, first);
        lastName = last;
        if (amt > 0)
            salary = amt;
        else
            salary = 0;
    }

    char* getFirstName() {
        return firstName;
    }
    string getLastName() {
        return lastName;
    }
    int getSalary() {
        return salary;
    }

    int getYearlySalary() {
        return salary * 12;
```

```
    }

    void raise(float perc) {
        salary += salary*perc/100.0;
    }
};

int main()
{
    Employee emp1, emp2;
    emp1.setInfo("Taha", "Ahmed", -500000);
    emp2.setInfo("Danial", "Ahmed", 500000);
    cout << "====Employee Info====" << endl;
    cout << endl << "Employee 1:" << endl << "Name: " <<
emp1.getFirstName() << " " << emp1.getLastName() << endl << "Yearly
Salary: " << emp1.getYearlySalary() << endl;
    cout << endl << "Employee 2:" << endl << "Name: " <<
emp2.getFirstName() << " " << emp2.getLastName() << endl << "Yearly
Salary: " << emp2.getYearlySalary() << endl;

    emp1.raise(10);
    emp2.raise(10);

    cout<<endl<< "After getting a raise:" << endl;
    cout << "====Employee Info====" << endl;
    cout << endl << "Employee 1:" << endl << "Name: " <<
emp1.getFirstName() << " " << emp1.getLastName() << endl << "Yearly
Salary: " << emp1.getYearlySalary() << endl;
    cout << endl << "Employee 2:" << endl << "Name: " <<
emp2.getFirstName() << " " << emp2.getLastName() << endl << "Yearly
Salary: " << emp2.getYearlySalary() << endl;
    return 0;
}
```

OUTPUT:

```
====Employee Info====

Employee 1:
Name: Taha Ahmed
Yearly Salary: 0

Employee 2:
Name: Danial Ahmed
Yearly Salary: 6000000

After getting a raise:
====Employee Info====

Employee 1:
Name: Taha Ahmed
Yearly Salary: 0

Employee 2:
Name: Danial Ahmed
Yearly Salary: 6600000
```

QUESTION#4:

CODE:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

class Game
{
public:
    int players, wins, losses;
    Game() : players(rand() % 10 + 1), wins(0), losses(0) {}

    void hit(void)
    {
        int num1 = rand() % 5 + 1;
        int num2 = rand() % 5 + 1;
        cout << "Pair of Numbers:" << endl;
        cout << "Number 1: " << num1 << endl;
        cout << "Number 2: " << num2 << endl;
        if (num1 == num2)
        {
            cout << "Enemy got hit by your team!" << endl;
            wins++;
        }
        else
        {
            cout << "You got hit by the enemy team!" << endl;
            losses++;
        }
    }

    void displayResult(void)
    {
        cout << "Game Over! ";
        if (wins > losses)
            cout << "You won" << endl;
        else if (losses > wins)
            cout << "You lost" << endl;
```

```
        else
            cout << "The game is a draw" << endl;
    }
};

int main(void)
{
    srand(time(0));
    Game team;
    cout << "Number of players in the team: " << team.players << endl;
    for (int i = 0; i < team.players; i++)
        team.hit();
    team.displayResult();
    return 0;
}
```

OUTPUT:

```
Number of players in the team: 5
Pair of Numbers:
Number 1: 5
Number 2: 1
You got hit by the enemy team!
Pair of Numbers:
Number 1: 2
Number 2: 2
Enemy got hit by your team!
Pair of Numbers:
Number 1: 2
Number 2: 5
You got hit by the enemy team!
Pair of Numbers:
Number 1: 4
Number 2: 3
You got hit by the enemy team!
Pair of Numbers:
Number 1: 4
Number 2: 4
Enemy got hit by your team!
Game Over! You lost
```