

# Practical-9

Date: \_\_\_\_\_

## AIM:

- i. Create a class MATRIX of size mxn. Overload + and - operators for addition and subtraction of the MATRIX.
- ii. Define a class Coord, which has x and y coordinates as its data members. Overload ++ and - operators for the Coord class. Create both its prefix and postfix forms.
- iii. Create one class called Rupees, which has one member data to store amount in rupee and create another class called Paise which has member data to store amount in paise. Write a program to convert one amount to another amount with use of type conversion.
- iv. Create two classes Celsius and Fahrenheit to store temperature in terms of Celsius and Fahrenheit respectively. Include necessary functions to read and display the values. Define conversion mechanism to convert Celsius object to Fahrenheit object and vice versa. Show both types of conversions in main function.

ii.

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

class Matrix {
private:
    int rows, cols;
    vector<vector<double>> data;
public:
    Matrix (int m, int n) :
        rows(m), cols(n), data (m, vector<double>(n)) {}
    void input() {
        cout << "Enter elements for a: " << rows << " x " << cols <<
            " matrix: \n";
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                cin >> data[i][j];
            }
        }
    }
    void display() const {
        cout << "Matrix (" << rows << " x " << cols << " ): \n";
        for (const auto& row : data) {
            for (const auto& elem : row) {
                cout << elem << " ";
            }
            cout << "\n";
        }
    }
}
```

```
Matrix operator+ (const Matrix & other) const {
    Matrix result (rows, cols);
    for (int i=0; i<rows; i++) {
        for (int j=0; j<cols; j++) {
            result.data[i][j] = data[i][j] + other.data[i][j];
        }
    }
    return result;
}
```

```
Matrix operator- (const Matrix & other) const {
    Matrix result (rows, cols);
    for (int i=0; i<rows; i++) {
        for (int j=0; j<cols; j++) {
            result.data[i][j] = data[i][j] - other.data[i][j];
        }
    }
    return result;
}
```

```
int main() {
    int m,n;
    cout << "Enter number of rows and columns for Matrix : ";
    cin >> m >> n;
    Matrix A(m,n);
    A.input();
    Matrix B(m,n);
    B.input();
    cout << "In Result of A+B : \n";
    Matrix C = A + B;
    C.display();
    cout << "In Result of A-B : \n";
    Matrix D = A - B;
    D.display();
    return 0;
}
```

ii.

```

#include <iostream>
using namespace std;

class coord {
private:
    int x, y;
public:
    coord (int xVal = 0, int yVal = 0) : x(xVal), y(yVal) {}
    void display() {
        cout << "Coordinates : (" << x << ", " << y << ")\n";
    }
    coord& operator++() {
        x++;
        y++;
        return *this;
    }
    coord operator++(int) {
        coord temp = *this;
        x++;
        y++;
        return temp;
    }
    coord& operator--() {
        x--;
        y--;
        return *this;
    }
    coord operator--(int) {
        coord temp = *this;
        x--;
        y--;
        return temp;
    }
};

```

```

int main() {
    int x, y;

    cout << "Enter initial x and y coordinates : ";
    cin >> x >> y;
    coord point(x, y);
    cout << "Original ";
    point.display();
    cout << "In Applying postfix ++ in ";
    point++;
    point.display();
    cout << "In Applying prefix -- in ";
    point--;
    point.display();
    cout << "In Applying postfix -- in ";
    point--;
    point.display();
    return 0;
}

```

Output :-

Enter initial x and y coordinates : 3 4  
 Original coordinates : (3, 4)

Applying prefix ++  
 (coordinates : (4, 5))  
 Applying postfix ++  
 (coordinates : (5, 6))  
 Applying prefix --  
 (coordinates : (4, 5))  
 Applying postfix --  
 (coordinates : (3, 4))

iii.

```

#include <iostream>
using namespace std;

class Paise;
class Rupees {
private:
    double rupees;
public:
    Rupees (double amount) : rupees(amount) {}
    double get_amount() const {
        return rupees;
    }
    operator Paise();
};

class Paise {
private:
    double paise;
public:
    Paise (double amount) : paise(amount) {}
    double get_amount() const {
        return paise;
    }
    operator Rupees();
};

Rupees :: operator Paise() {
    return Paise(rupees * 100);
}

Paise :: operator Rupees() {
    return Rupees(paise / 100);
}

int main {
    int a;
    double amount;
}

```

```
cout << "Select Conversion Method : In 1: Rupees → Paisa \n"
        2. Paisa → Rupees \n\n";
cin >> a;

switch (case(a)) {
    case 1:
    {
        cout << "Rupees → Paisa Selected : \n";
        cout << "Enter your Rupee amount : ";
        cin >> amount;

        Rupees rupees(amount);
        Paisa paise = rupees;
        cout << "Converted Amount : " << paise.get_amount() <<
                "paise" << endl;
        break;
    }

    case 2:
    {
        cout << "Paisa → Rupees Selected : \n";
        cout << "Enter your Paisa amount : ";
        cin >> amount;

        Paisa paise(amount);
        Rupees rupees = paise;
        cout << "Converted Amount : " << rupees.get_amount() <<
                "rupees" << endl;
        break;
    }

    default:
        cout << "Invalid Input \n";
        return 1;
}

return 0;
```

Output :-

Select conversion Method :

1. Rupees → Paise
2. Paise → Rupees

1

Rupees → Paise Selected :

Enter your Rupee amount : 5348  
Converted amount : 534800 paise