

## PRACTICAL-5

DATE: \_\_\_\_\_

### **AIM:-**

- i. Using friend function find the maximum number from given two To learn how numbers from two different classes. Write all necessary functions and to implement constructors for the program.
- ii. Constructors Using a friend function, find the average of three numbers from three different classes. Write all necessary member functions and constructor for the classes. and class member functions.
- iii. Define currency class which contains rupees and paisa as data members. Write a friend function named AddCurrency() which add 2 different Currency objects and returns a Currency object. Write parameterized constructor to initialize the values and use appropriate functions to get the details from the user and display it.
- iv. Create Calendar class with day, month and year as data members. Include default and parameterized constructors to initialize a Calendar object with a valid date value. Define a function AddDays to add days to the Calendar object. Define a display function to show data in "dd/mm/yyyy" format.

### INPUT:-

ij.

```
#include <iostream>
using namespace std;
class B;
class A {
private :
    int numA;
public :
    A(int a) : numA(a) {}
    friend int findMax(A, B);
};

class B {
private :
    int numB;
public :
    B(int b) : numB(b) {}
    friend int findMax(A, B);
};

int findMax (A objA, B objB) {
```

```

if (objA.numA > objB.numB) {
    return objA.numA;
} else if {
    return objB.numB;
}
int main() {
    A objA(10);
    B objB(20);
    int maxNumber = findMax(objA, objB);
    cout << "The maximum number is " << maxNumber <<
        endl;
    return 0;
}

```

Output :-

The maximum number is 20.

ii).

```

#include <iostream>
using namespace std;
class B;
class C;
class A {
private:
    int numA;
public:
    A(int a) : num(a) {}
    friend float findAverage(A, B, C);
};
class B {
private:
    int numB;
public:

```

```

B (int b) : numB(b) {}
friend float findAverage (A, B, C);
};

class C {
private :
    int numC;
public :
    (int c) : numC (c) {}
friend float findAverage (A, B, C);
};

float findAverage (A objA, B objB, C objC) {
    float total = objA.numA + objB.numB + objC.numC;
    return total / 3;
}

int main () {
    A objA (10);
    B objB (20);
    C objC (30);
    float avg = findAverage (objA, objB, objC);
    cout << "The average of Three numbers is " << avg << endl;
    return 0;
}

```

Output:-

The average of Three numbers is 20

iii).

```

#include <
using namespace std;

class Currency {
private :

```

```

int rupees;
int paisa;
public :
    Currency (int r=0, int p=0) {
        rupees = r;
        paisa = p;
        normalize();
    }
    friend Currency AddCurrency (const Currency&, const
                                Currency&);
    void getDetails() {
        cout << "Enter rupees : ";
        cin >> rupees;
        cout << "Enter paisa : ";
        cin >> paisa;
        normalize();
    }
    void display() const {
        cout << "Currency : " << rupees << "Rupees and " << paisa;
    }
private :
    void normalize() {
        if (paisa >= 100) {
            rupees += paisa / 100;
            paisa = paisa % 100;
        }
    }
};

int main() {
    Currency c1, c2;
    cout << "Enter details for first currency : \n";
    c1.getDetails();
    cout << "Enter details for second currency : \n";
    c2.getDetails();
    Currency result = AddCurrency (c1, c2);
    result.display();
}

```

Output :-

Enter details for first currency :

Enter rupees : 5

Enter paisa : 75

Enter details for second currency :

Enter rupees : 3

Enter paisa : 50

After adding the two currencies :

Currency : 9 rupees and 25 paisa

iv).

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

class Calendar {
    int date, month, year;
public :
    Calendar() : date(0), month(0), year(0) {}
    Calendar(int a, int b, int c) : date(a), month(b), year(c) {}
    bool isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400
                                                       == 0);
    }
    void add() {
        int a;
        cout << "Enter the number of days you want to add : ";
        cin >> a;
        date += a;
    }
    do {
        if ((month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) && date > 31) {
```

```

date -= 31;
month++;

}

else if ((month == 4 || month == 6 || month == 9 || month == 11) &&
         date > 30) {
    date -= 30;
    month++;
}

else if (month == 2) {
    int maxDays = isLeapYear(year) ? 29 : 28;
    if (date > maxDays) {
        date -= maxDays;
        month++;
    }
}

if (month > 12) {
    month -= 12;
    year++;
}

} while (date > 31 || month > 12);
}

void display() {
    cout << (date < 10 ? "0" : "") << date << '/'
        << (month < 10 ? "0" : "") << month << '/'
        << year << endl;
}

int main() {
    int a, b, c;
    cout << "Enter DATE : ";
    cin >> a;
    cout << "Enter MONTH : ";
    cin >> b;
    cout << "Enter YEAR : ";
    cin >> c;
}

```

```
(calendar obj1(a, b, c);  
obj1.add();  
obj1.display();  
return 0;  
y
```

Output :-

Enter Date : 15

Enter Month : 10

Enter Year : 2024

Enter the number of days you want to add : 50

04/12/2024