Subject:- C++
LAB Assignment - 4

1. **Write a C++ program to accept and display two data members x and y which performs SUM operation of two objects using objects as Argument.**
   Like
   obj1.x = 10
   obj2.x = 30
   and obj1.y= 20
   and obj2.y =40
   sum of two objects
   obj3.x = 40
   and obj3.y= 60

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

// Class definition
class Number {
public:
        int x, y;  // Data members

        // Function to accept values for x and y
        void accept(int a, int b) {
        x = a;
        y = b;
        }

        // Function to sum the data members of two objects
        Number sum(Number obj) {
        Number result;  // Create a new object to store the sum
        result.x = x + obj.x;
        result.y = y + obj.y;
```

```
        return result;  // Return the result object
        }

        // Function to display the values of x and y
        void display() {
        cout << "x = " << x << ", y = " << y << endl;
        }
};

int main() {
        Number obj1, obj2, obj3;  // Create three objects

        // Accept values for obj1 and obj2
        obj1.accept(10, 20);
        obj2.accept(30, 40);

        // Perform the sum of obj1 and obj2 and store in obj3
        obj3 = obj1.sum(obj2);

        // Display the values of all objects
        cout << "Values of obj1: ";
        obj1.display();
        cout << "Values of obj2: ";
        obj2.display();
        cout << "Sum of obj1 and obj2 (obj3): ";
        obj3.display();

        return 0;
}
```

Output:

```
Enter values for obj1 (x and y): 10 20
Enter values for obj2 (x and y): 30 40
Object 1: x: 10, y: 20
Object 2: x: 30, y: 40
Sum of objects: x: 40, y: 60
```

2. **Write a C++ program that performs sum with passing two objects argument and return also object as argument.**

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

// Class definition
class Number {
public:
        int x, y;  // Data members

        // Function to accept values for x and y
        void accept(int a, int b) {
        x = a;
        y = b;
        }

        // Function to display values of x and y
        void display() {
        cout << "x = " << x << ", y = " << y << endl;
        }

        // Friend function to sum two objects and return the result as
an     object
        friend Number sum(Number obj1, Number obj2);
};

// Function to sum the data members of two objects and return the result
as an object
Number sum(Number obj1, Number obj2) {
        Number result;  // Create a new object to store the sum
        result.x = obj1.x + obj2.x;
        result.y = obj1.y + obj2.y;
        return result;  // Return the result object
}

int main() {
```

```
    Number obj1, obj2, obj3;  // Create three objects

    // Accept values for obj1 and obj2
    obj1.accept(10, 20);
    obj2.accept(30, 40);

    // Call the sum function by passing two objects and store the result
 in obj3
    obj3 = sum(obj1, obj2);

    // Display the values of all objects
    cout << "Values of obj1: ";
    obj1.display();
    cout << "Values of obj2: ";
    obj2.display();
    cout << "Sum of obj1 and obj2 (obj3): ";
    obj3.display();

    return 0;
}
```

Output:

```
yaml                                                    Copy code

Enter values for obj1 (x and y): 10 20
Enter values for obj2 (x and y): 30 40
Object 1: x: 10, y: 20
Object 2: x: 30, y: 40
Sum of objects: x: 40, y: 60
```

3. **Define a class to represent a bank account.Include the following members:**
   Data member
   a. Name of the depositor
   b. Account Number
   c. Type of account
   d. Balance amount in the account
   Member Functions

a. To assign initial values
b. To deposit an amount
c. To withdraw an amount after checking the balance
d. To display name and balance.
Write a main program to test the program.

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

// Class definition
class BankAccount {
private:
    string depositorName;   // Name of the depositor
    int accountNumber;      // Account number
    string accountType;     // Type of account (e.g., Savings,
Current)
    double balance;         // Balance amount in the account

public:
    // Function to assign initial values
    void assignInitialValues(string name, int accNo, string type,
double bal) {
    depositorName = name;
    accountNumber = accNo;
    accountType = type;
    balance = bal;
    }

    // Function to deposit an amount
    void depositAmount(double amount) {
    balance += amount;
    cout << "Deposited: " << amount << ". Updated Balance: " <<
balance << endl;
    }

    // Function to withdraw an amount after checking the balance
```

```
        void withdrawAmount(double amount) {
        if (amount <= balance) {
        balance -= amount;
        cout << "Withdrawn: " << amount << ". Updated Balance: " <<
balance << endl;
        } else {
        cout << "Insufficient balance. Withdrawal failed!" << endl;
        }
        }

        // Function to display the name and balance
        void displayAccountInfo() {
        cout << "Depositor Name: " << depositorName << endl;
        cout << "Account Number: " << accountNumber << endl;
        cout << "Account Type: " << accountType << endl;
        cout << "Current Balance: " << balance << endl;
        }
    };

    int main() {
        BankAccount account;  // Create an object of BankAccount

        // Assign initial values to the bank account
        account.assignInitialValues("Jayshree Donga", 123456789,
"Savings", 5000.0);

        // Display the account information
        account.displayAccountInfo();

        // Deposit an amount
        account.depositAmount(2000.0);

        // Try to withdraw an amount
        account.withdrawAmount(3000.0);

        // Try to withdraw an amount larger than the balance
        account.withdrawAmount(5000.0);
```

```
        // Display the updated account information
        account.displayAccountInfo();

        return 0;
}
```

Output: