```cpp
/**
 * Lab Sheet - 1: Data Structure and Algorithm
 * Title: Authentication key representation for the multiple Users in CPP.
 * Aim: Utilization of Union/Structure/Enum type representation using inheritance.
 * author: Santa Basnet
 * Everest Engineering College.
 */

#include <iostream>
#include <sstream>
#include <utility>

#define MAX 5

using namespace std;

class SoftwareUser {
protected:
    int id;
    string name;
    int price;
public:
    SoftwareUser(int id, string name, int price) {
        this->id = id;
        this->name = std::move(name);
        this->price = price;
    }

    virtual void display() = 0;

    virtual int numberOfLicenses() = 0;

    virtual string userType() = 0;

    virtual int amount() {
        return this->price;
    };

    virtual string getName() {
        return this->name;
    }

    virtual bool hasKey(string) = 0;
};

class IndividualUser : public virtual SoftwareUser {
    string licenseKey;
public:
    IndividualUser(int id, string name, string licenseKey, int price) :
SoftwareUser(id, std::move(name), price) {
        this->licenseKey = std::move(licenseKey);
    }

    int numberOfLicenses() override {
        return 1;
    }

    string userType() override {
        return "INDIVIDUAL";
    }

    void display() override {
        cout << endl;
        cout << this->id << "\t"
             << this->name << "\t"
             << userType() << "\t"
             << this->licenseKey << "\t"
```

```cpp
                << this->price << "\t";
    }

    bool hasKey(string givenKey) override {
        return this->licenseKey == givenKey;
    }

};

class CorporateUser : public virtual SoftwareUser {
    int noOfLicenseKeys;
    string licenseKeys[25];

    string formatLicenseKeys() {
        std::ostringstream buffer;
        for (int i = 0; i < this->noOfLicenseKeys; i++) buffer << licenseKeys[i] << ",
";
        return buffer.str();
    }

public:
    CorporateUser(int id, string name, string *licenseKeys, int price, int
noOfLicenseKeys)
            : SoftwareUser(id, std::move(name), price) {
        this->noOfLicenseKeys = noOfLicenseKeys;
        std::copy(licenseKeys, licenseKeys + noOfLicenseKeys, this->licenseKeys);
    }

    int numberOfLicenses() override {
        return this->noOfLicenseKeys;
    }

    string userType() override {
        return "CORPORATE";
    }

    void display() override {
        cout << endl;
        cout << this->id << "\t"
             << this->name << "\t"
             << userType() << "\t"
             << this->formatLicenseKeys() << "\t"
             << this->price << "\t"
             << this->noOfLicenseKeys;
    }

    bool hasKey(string givenKey) override {
        for (auto &key: this->licenseKeys) {
            if (key == givenKey) return true;
        };
        return false;
    }
};

/**
 * Global variable represents all the users.
 */

SoftwareUser *allUsers[MAX];

void initialize() {
    string licenses1[] = {"by3K", "66Y2", "4Uee", "oQ5r"};
    string licenses2[] = {"Lrc4", "rc8K", "22Y8"};
    string licenses3[] = {"8tW2", "uU5e", "pPi0", "eR71", "Tup4", "v9Bg"};

    allUsers[0] = new IndividualUser(1, "ram", "aXz5", 5000);
    allUsers[1] = new CorporateUser(2, "eec", licenses1, 15000, 4);
    allUsers[2] = new CorporateUser(3, "pec", licenses2, 10000, 3);
    allUsers[3] = new CorporateUser(4, "kec", licenses3, 25000, 6);
    allUsers[4] = new IndividualUser(5, "sam", "aXz5", 5000);
}

/**
 * Display All Users.
 */
void displayAllUsers() {
    cout << endl << "2. All the listed Users : ";
    for (auto &allUser: allUsers) allUser->display();
}

int amountOf(string userName) {
    for (auto &user: allUsers) {
        if (user->getName() == userName) return user->amount();
    }
    return -1;
}

string userTypeOf(string userName) {
    for (auto &user: allUsers) {
        if (user->getName() == userName) return user->userType();
    }
    return "Not Found !";
}

string findAccessOf(string userName, string licenseKey) {
    bool result = false;
    for (auto &user: allUsers) {
        if (user->getName() == userName && user->hasKey(licenseKey)) {
            result = true;
            break;
        };
    }
    if (result) return "Access Granted.";
    else return "Access Denied !";
}

/**
 * Display Summary.
 */
void displaySummary() {
    int noOfUsers = 0;
    int noOfLicenses = 0;
    int totalAmountCollected = 0;
    for (auto &user: allUsers) {
        noOfUsers++;
        noOfLicenses += user->numberOfLicenses();
        totalAmountCollected += user->amount();
    }
    cout << endl << "6. Summaries";
    cout << endl << "\t" << "No. of Users: " << noOfUsers;
    cout << endl << "\t" << "No. of Licenses Given: " << noOfLicenses;
    cout << endl << "\t" << "total Amount Collected: " << totalAmountCollected;
}

/**
 * Program entry point.
 * @return 0 for successful operation.
 */
int main() {
    /**
     * 1. Initialize the table data.
     */
    initialize();
    /**
     * 1. Display all the initialized data.
     */
    displayAllUsers();

    /**
     * 2. Find the price paid by the user name.
     */
    cout << endl;
    string userName("kec");
    cout << endl << "3. Price paid by the user : " << amountOf(userName) << endl;

    /**
     * 3. Find the user type by the user name.
     */
    cout << endl << "4. User type of: " << userTypeOf(userName) << endl;

    /**
     * 4. Find the access key by given the user name and the user key.
     */
    string licenseKey("pPi0");
    cout << endl << "5. License key response: " << findAccessOf(userName, licenseKey)
<< endl;

    /**
     * 5. Display Summary.
     */
    displaySummary();
    cout << endl;

    return 0;
}

Output:
"C:\Work\Lectures\EEC\Data Structures and Algorithm\Course\main.exe"

2. All the listed Users :
1       ram     INDIVIDUAL      aXz5    5000
2       eec     CORPORATE       by3K, 66Y2, 4Uee, oQ5r,         15000   4
3       pec     CORPORATE       Lrc4, rc8K, 22Y8,       10000   3
4       kec     CORPORATE       8tW2, uU5e, pPi0, eR71, Tup4, v9Bg,     25000   6
5       sam     INDIVIDUAL      aXz5    5000

3. Price paid by the user : 25000

4. User type of: CORPORATE

5. License key response: Access Granted.

6. Summaries
        No. of Users: 5
        No. of Licenses Given: 15
        Total Amount Collected: 60000

Process finished with exit code 0
```