```cpp
#include <iostream>
#include <string>
#include <iomanip>
#include <cmath>
#include <stdexcept>

using namespace std;

class Mobile
{
public:
    string brand;
    string model;
    double price;
    int stock;
    bool emiAvailable;
};

class Buyer
{
public:
    string name;
    string mobileNumber;
    string address;
    string dateOfPurchase;
    string paymentType; // "emi" or "cash"
    double downPayment;
    int remainingMonths;
};

class Shop
{
private:
    static const int MAX_INVENTORY_SIZE = 100;
    static const int MAX_BUYERS_SIZE = 100;

    Mobile inventory[MAX_INVENTORY_SIZE];
    Buyer buyers[MAX_BUYERS_SIZE];
    int inventorySize = 0;
    int buyersSize = 0;

public:
    // Function to add a mobile phone to the inventory
    void addMobilePhone()
    {
        if (inventorySize < MAX_INVENTORY_SIZE)
        {
            Mobile mobile;
```

```
            cout << "Enter brand: ";
            cin >> mobile.brand;
            cout << "Enter model: ";
            cin >> mobile.model;
            cout << "Enter price: ";
            cin >> mobile.price;
            cout << "Enter stock quantity: ";
            cin >> mobile.stock;

            mobile.emiAvailable = (mobile.price > 15000);
            inventory[inventorySize++] = mobile;
            cout << "Mobile phone added to inventory." << endl;
            cout<<"-----------------------------------------------------
"<<endl;
        }
        else
        {
            cout << "Inventory is full. Cannot add more mobile phones." <<
endl;
                        cout<<"-----------------------------------------------
-------"<<endl;
        }
    }

    // Function to sell a mobile phone
    void sellMobilePhone()
    {
        if (inventorySize == 0)
        {
            cout << "Inventory is empty. Cannot sell mobile phones." << endl;
            return;
        }

        string sellBrand, sellModel;
        bool mobileFound = false;

        do
        {
            cout << "Enter brand of the mobile: ";
            cin >> sellBrand;
            cout << "Enter the model to sell: ";
            cin >> sellModel;

            for (int i = 0; i < inventorySize; i++)
            {
                if (inventory[i].brand == sellBrand && inventory[i].model ==
sellModel)
                {
```

```
                    if (inventory[i].stock > 0)
                    {
                        Buyer buyer;
                        cout << "Enter buyer's name: ";
                        cin >> buyer.name;
                        cout << "Enter buyer's mobile number: ";
                        cin >> buyer.mobileNumber;
                        cout << "Enter buyer's address: ";
                        cin >> buyer.address;
                        cout << "Enter date of purchase: ";
                        cin >> buyer.dateOfPurchase;
                        cout << "Select payment type (emi/cash): ";
                        cin >> buyer.paymentType;

                        if (buyer.paymentType == "emi" &&
inventory[i].emiAvailable)
                        {
                            buyer.downPayment = 0.3 * inventory[i].price;
                            cout << " Down payment will be " <<
buyer.downPayment << "INR"
                                 << " \n Do you want to pay the remaining
amount in  less than 8 months? (yes/no): ";
                            string choice;
                            cin >> choice;

                            if (choice == "yes")
                            {
                                cout << "Enter the number of months to pay off
the remaining amount (max 8 months): ";
                                cin >> buyer.remainingMonths;

                                if (buyer.remainingMonths > 8)
                                {
                                    buyer.remainingMonths = 8;
                                }
                            }
                            else
                            {
                                buyer.remainingMonths = 8; // Default to 8
months if not paying in less than 8 months.
                            }

                            double monthlyPayment = (inventory[i].price -
buyer.downPayment) / buyer.remainingMonths;

                            cout << "\nDown Payment: " << buyer.downPayment <<
" INR" << endl;
```

```
                            cout << "Amount Payable per Month: " <<
monthlyPayment << " INR" << endl;
                            cout << "Number of Months: " <<
buyer.remainingMonths << " months" << endl;
                            cout<<endl;
                        }
                        else if (buyer.paymentType == "cash")
                        {
                            buyer.downPayment = inventory[i].price;
                            buyer.remainingMonths = 0;
                            cout << "\nDown Payment: " << buyer.downPayment <<
" INR" << endl;
                            cout << "Amount Payable per Month: N/A (Cash
payment)" << endl;
                            cout << "Number of Months: N/A (Cash payment)" <<
endl;
                            cout<<endl;
                        }
                        else
                        {
                            throw runtime_error("Invalid payment type or EMI
not available for this mobile.");
                        }

                        buyers[buyersSize++] = buyer;
                        inventory[i].stock--;
                        mobileFound = true;
                        generateReceipt(buyer, inventory[i]); // Generate and
display the receipt
                    }
                    else
                    {
                        throw runtime_error("Out of stock!");
                    }
                }
            }

            if (!mobileFound)
            {
                cout << "Mobile phone not found in inventory. Please try
again." << endl;
            }

        } while (!mobileFound);
    }

    // Function to display mobile phones in inventory
    void displayMobilePhones()
```

```
    {
        cout <<
"\n*****************************************************************************
*********************************************" << endl;
        cout << "                                                Mobile Phones in
Inventory" << endl;
        cout <<
"*****************************************************************************
*********************************************" << endl;
        cout << left << setw(15) << "Brand" << setw(15) << "Model" << setw(15)
<< "Price (INR)" << setw(15) << "Stock" << setw(15) << "EMI Available" <<
endl;
        cout << "-----------------------------------------------------------
-----------------------------------------------------------" << endl;
        for (int i = 0; i < inventorySize; i++)
        {
            cout << left << setw(15) << inventory[i].brand << setw(15) <<
inventory[i].model
                 << setw(15) << inventory[i].price << setw(15) <<
inventory[i].stock
                 << setw(15) << (inventory[i].emiAvailable ? "Yes" : "No") <<
endl;
        }
        cout <<
"*****************************************************************************
*********************************************" << endl;
        cout<<endl;
        cout<<"-----------------------------------------------------"<<endl;
    }

    // Function to check units available per brand and model
    void checkUnitsAvailable()
    {
        string brand, model;
        cout << "\nEnter brand name: ";
        cin >> brand;
        cout << "Enter model: ";
        cin >> model;

        int unitsAvailable = 0;
        for (int i = 0; i < inventorySize; i++)
        {
            if (inventory[i].brand == brand && inventory[i].model == model)
            {
                unitsAvailable = inventory[i].stock;
                break;
            }
        }
```

```
        cout << "Units available for " << brand << " " << model << ": " <<
unitsAvailable << endl;
                    cout<<"-------------------------------------------------
---"<<endl;
    }

    // Function to display buyer details
    void displayBuyerDetails()
    {
        cout <<
"\n*********************************************************************
*********************************************************************
*********************************************" << endl;
        cout <<
"                                                                    Buyer
Details" << endl;
        cout <<
"\n*********************************************************************
*********************************************************************
*********************************************" << endl;
        cout << left << setw(15) << "Name" << setw(15) << "Mobile Number" <<
setw(15) << "Address" << setw(15) << "Date of Purchase" << setw(15) <<
"Payment Type" << setw(20) << "Down Payment (INR)" << setw(25) << "Amount
Payable per Month (INR)" << setw(20) << "Number of Months" << endl;
        cout << "-------------------------------------------------------------
-----------------------------------------------------------------------------
-------------------------------------------------------------" << endl;
        for (int i = 0; i < buyersSize; i++)
        {
            cout << left << setw(15) << buyers[i].name << setw(15) <<
buyers[i].mobileNumber
                 << setw(15) << buyers[i].address << setw(15) <<
buyers[i].dateOfPurchase
                 << setw(15) << buyers[i].paymentType << setw(20) <<
buyers[i].downPayment;
            if (buyers[i].paymentType == "emi")
            {
                cout << setw(25) << (buyers[i].remainingMonths > 0 ?
(inventory[0].price - buyers[i].downPayment) / buyers[i].remainingMonths : 0);
                cout << setw(20) << buyers[i].remainingMonths;
            }
            else
            {
                cout << setw(25) << "N/A (Cash payment)";
                cout << setw(20) << "N/A";
            }
            cout << endl;
```

```
        }
       cout <<
"\n**********************************************************************
**********************************************************************
**********************************************" << endl;

        cout << "\n1. Main Menu" << endl;
        cout << "2. Exit" << endl;
        int choice;
        cout << "Enter your choice: ";
        cin >> choice;
                    cout<<"-----------------------------------------------
---"<<endl;
        if (choice == 1)
        {
            // Continue to the main menu
        }
        else if (choice == 2)
        {
            cout << "THANK YOU" << endl;
            exit(0);
        }
        else
        {
            cout << "Invalid choice. Returning to the main menu." << endl;
        }
    }
    void generateReceipt(const Buyer &buyer, const Mobile &mobile)
    {
        cout << "****************************************" << endl;
        cout << "Receipt" << endl;
        cout << "****************************************" << endl;
        cout << "Buyer Name: " << buyer.name << endl;
        cout << "Buyer Mobile Number: " << buyer.mobileNumber << endl;
        cout << "Buyer Address: " << buyer.address << endl;
        cout << "Date of Purchase: " << buyer.dateOfPurchase << endl;
        cout << "Brand: " << mobile.brand << endl;
        cout << "Model: " << mobile.model << endl;
        cout << "Price (INR): " << mobile.price << endl;
        cout << "Payment Type: " << buyer.paymentType << endl;
        cout << "Down Payment (INR): " << buyer.downPayment << endl;

        if (buyer.paymentType == "emi")
        {
            cout << "Amount Payable per Month (INR): " << (mobile.price -
buyer.downPayment) / buyer.remainingMonths << endl;
            cout << "Number of Months: " << buyer.remainingMonths << " months"
<< endl;
```

```
        }
        else
        {
            cout << "Amount Payable per Month: N/A (Cash payment)" << endl;
            cout << "Number of Months: N/A (Cash payment)" << endl;

        }
        cout<<endl;
        cout << "****************************************" << endl;
        cout<<endl;
        cout<<"-------------------------------------------------"<<endl;
    }

    // Function to display the main menu
    void displayMainMenu()
    {

        cout<<"Enter the operation you want to perform"<<endl;
        cout<<"-------------------------------------------------"<<endl;

        cout << "1. Add Mobile Phone" << endl;
        cout << "2. Sell Mobile Phone" << endl;
        cout << "3. Display Mobile Phones" << endl;
        cout << "4. Check Units Available per Brand and Model" << endl;
        cout << "5. Display Buyer Details" << endl;
        cout << "6. Exit" << endl;
    }
};

int main()
{
    Shop shop;
        cout << " _____________________________________________" << endl;
        cout << " |      Mobile Shop Management System        | " << endl;
        cout << " |___________________________________________|" << endl;
    while (true)
    {
        shop.displayMainMenu();
        int choice;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            shop.addMobilePhone();
            break;
```

```
        case 2:
            shop.sellMobilePhone();
            break;

        case 3:
            shop.displayMobilePhones();
            break;

        case 4:
            shop.checkUnitsAvailable();
            break;

        case 5:
            shop.displayBuyerDetails();
            break;

        case 6:
            cout << "THANK YOU" << endl;
            return 0;

        default:
            cout << "Invalid choice. Please try again." << endl;
        }
    }

    return 0;
}
```