Problem 1: Library Management System

```
#include <bits/stdc++.h>
using namespace std;

class Book {

private:
    string title;
    string author;
    string ISBN;

public:

    Book(string title1, string author1, string ISBN1, int x) : title(title1), author(author1),
ISBN(ISBN1) {
        cout << "Book " << x << " created" << endl;
    }

    ~Book() {
        cout << "destroyed" << endl;
    }
};

int main() {

    Book *b1 = new Book("Introduction to OOP", "John Doe", "123456789", 1);
    Book *b2 = new Book("Data Structures and Algorithms", "Jane Smith", "987654321", 2);

    int y = 1;
    cout << "Book " << y << " ";
    delete b1;
    int z = 2;
    cout << "Book " << z << " ";
    delete b2;
}
```

Problem 2: Mkash Account Management

```
#include <bits/stdc++.h>
using namespace std;

class Account {
public:
    int accountNumber;
    string accountHolder;
    int balance;

    Account(int accountNo2, string accountH2, int blnc2) {
        accountNumber = accountNo2;
        accountHolder = accountH2;
        balance = blnc2;
        cout << "Account 2 created with balance " << balance << endl;  // Account 2
created with balance 5000
    }

    Account(int accountNo1, string accountH1) {
        accountNumber = accountNo1;
        accountHolder = accountH1;
        balance = 0;
        cout << "Accoutn 1 created with balance " << balance << endl;  // Account 1
created with balance 0
    }

    void deposit(int amount, int p) {
        balance = balance + amount;
        cout << "Deposit: " << p << ": " << amount << ". " << "New Balance: " << balance
<< endl; // Deposit     1: 1500. New Balance: 1500  //  Deposit 2: 400. New Balance: 5400
    }

    void withdrawal(int amount, int q) {
        balance = balance - amount;
        cout << "Withdrawal " << q << ": " << amount << ". " << "New Balance: " <<
balance << endl << endl; // Withdrawal 1: 800. New Balance: 700  // Withdrawal 2: 600. New
Balance: 4800
    }
};


int main() {
    Account a1(1001, "Alice");
    Account a2(1002, "Bob", 5000);
    a1.deposit(1500, 1);
    a1.withdrawal(800, 1);

    a2.deposit(400, 2);
    a2.withdrawal(600, 2);

    return 0;
}
```

Problem 3: Student Information System

```
#include <bits/stdc++.h>
using namespace std;

class Student {
public:
    int rollNumber;
    string name;
    int marks;

    Student(int rollNumber, string name, int marks) {
        this->rollNumber = rollNumber;
        this->name = name;
        this->marks = marks;
        cout << "Student 2 created with marks " << this->marks << ". ";  // Student 2
created with marks 75.
        if (70 <= (this->marks) < 80)
            cout << "Grade: " << "B" << endl;  //  Grade: B
    }

    Student(int rollNumber, string name) {
        this->rollNumber = rollNumber;
        this->name = name;
        cout << "Student 1 created with no marks" << endl;    // Student 1 created with no
marks
    }

};
```

```
int main() {
        Student s1(101, "Tom");
        Student s2(102, "Lisa", 75);
        return 0;
}
```

Problem 4: Car Rental System

```
#include <bits/stdc++.h>
using namespace std;

class Car {
public:
        string model;
        string make;
        int rentalFee;

        Car(string model, string make, int rentalFee) {
                this->model = model;
                this->make = make;
                this->rentalFee = rentalFee;
                cout << "Car 2 created with rental fee " << rentalFee << ". ";

        }

        Car(string model, string make) {
                this->model = model;
                this->make = make;
                cout << "Car 1 created with no rental fee" << endl;
        }

        void detail() {
                cout << "Rental Fee: " << rentalFee << endl;
        }

};


int main() {
        Car c1("Sedan", "Toyota");
        Car c2("SUV", "Ford", 50);
        c2.detail();

        return 0;
}
```

Problem 5: Employee Management System

```
#include <bits/stdc++.h>
using namespace std;

class Employee {
public:
        int employeeID;
        string name;
        int salary;

        Employee(int employeeID, string name, int salary) {
                this->employeeID = employeeID;
                this->name = name;
                this->salary = salary;
                cout << "Employee 2 created with salary " << salary << ". ";
        }

        Employee(int employeeID, string name) {
                this->employeeID = employeeID;
                this->name = name;
                cout << "Employee 1 created with no salary" << endl;
        }

        void detail() {
                int annualSalary = salary * 12;
                cout << "Annual Salary: " << annualSalary << endl;
        }
};

int main() {
        Employee c1(001, "John");
        Employee c2(002, "Jane", 5000);
        c2.detail();
}
```