Topic : Online Land Sales System

Group no : MLB_09.02_06

Campus : Malabe

Submission Date : 17/05/2022

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21177514	M.R. Tharin Ransika	0711758782
IT21178504	K.V.S.D. Dasanya	0711870278
IT21178740	K.K.L. Rathnasiri	0765809877
IT21179976	D.H.K. Hasini Ishara	0711909733

Approach

The assignment topic of our group received is Online Land Sales System. We proposed a name for the online land sales system. The name we suggested to online land sales system is **Dream Lands**.

Initially, we studied typical online land sales systems available on internet and **identified requirements** related to online land sales system. After we done **Object Oriented Analysis (utilizing Noun Verb Analysis method and CRC Cards)** to identify classes and relationships between classes.

After completing object oriented analysis, we done **Object Oriented Design**. At that phase we create Class Diagram.

In Implementation phase we implement classes and objects using C++ Programming Language.

Requirements Related to System

1. Dream Lands is an Online Land Sales Company. Which sales properties through the company's website.
2. Two types of customers interact with the website. They are **Registered Customers** and **Unregistered Customers**.
3. Registered customers have registered in the system by providing **Name**, **Permanent Address**, **Email Address**, **Contact Number**, and **NIC** to the system.
4. Registered customers have a **unique Customer ID Number**.
5. Registered customers **can visit details of properties** and **reserve properties** through the system. But unregistered customers **cannot reserve properties**. They can only **visit details of properties** through the website.
6. Company sales **three types of properties** through the system. They are **Lands**, **Apartments**, and **Homes**.
7. Each property has a **unique Property ID Number**.
8. The website displays **Property ID Number**, **Property Type**, **Province** particular property situated, a **Town** close to particular property, **Price**, **Status** (Whether reserved or not), and other features.
9. **One registered customer can reserve more than one property at a time but one property can be reserved by only one registered customer at a time**.
10. **System Admin, adds (Uploads) new properties** to the website, **updates details of properties** and **removes sold properties** from the website.
11. End of every month the **Manager of Dream Lands Company** receives the **Monthly Report of Properties** (which shows data on property sales).

Noun Verb Analysis

	Noun	Class	Rules of Rejecting Nouns
1.	Company		Outside scope of the system
2.	Properties	Class	
3.	Registered Customers	Class	
4.	Unregistered Customers		Outside scope of the system
5.	Name		Attribute
6.	Permanent Address		Attribute
7.	Email Address		Attribute
8.	Contact Number		Attribute
9.	NIC		Attribute
10.	Customer ID Number		Attribute
11.	Lands	Class	
12.	Apartments	Class	
13.	Homes	Class	
14.	Property ID Number		Attribute
15.	Property Type		Attribute
16.	Province		Attribute
17.	Town		Attribute
18.	Price		Attribute
19.	Status		Attribute
20.	System Admin		Outside scope of the system
21.	Manager		Outside scope of the system
22.	Monthly Report of Properties		Redundant

Assumptions Made in Noun Verb Analysis Process:

1. Unregistered customers, System admin and Manager are actors of the system. But they don't provide his/her data directly relevant to the system domain. Therefore, we didn't consider Unregistered customers, System admin and Manager as classes.
2. Property class represents attributes and methods common for Land class, Home class and Apartment class. To represent attributes and methods unique for Lands, Homes and Apartments, we created Land class, Home class and Apartment class.

    Ex:

    1. Number of rooms in Home or an Apartment
    2. Size (Area) of Land

CRC Cards

Registered Customer Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	

Property Class	
Responsibilities	**Collaborations**
Set Details	Land, Home, Apartment
Update Details	Land, Home, Apartment
Display Details	Land, Home, Apartment
Remove Details	Land, Home, Apartment
Reserve	Registered Customer, Land, Home, Apartment
Report of Properties	Land, Home, Apartment

Home Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Land Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Apartment Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Refine CRC Cards

Registered Customer Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	

Property Class	
Responsibilities	**Collaborations**
Set Details	Land, Home, Apartment
Update Details	Land, Home, Apartment
Display Details	Land, Home, Apartment
Remove Details	Land, Home, Apartment
Reserve	Registered Customer, Land, Home, Apartment

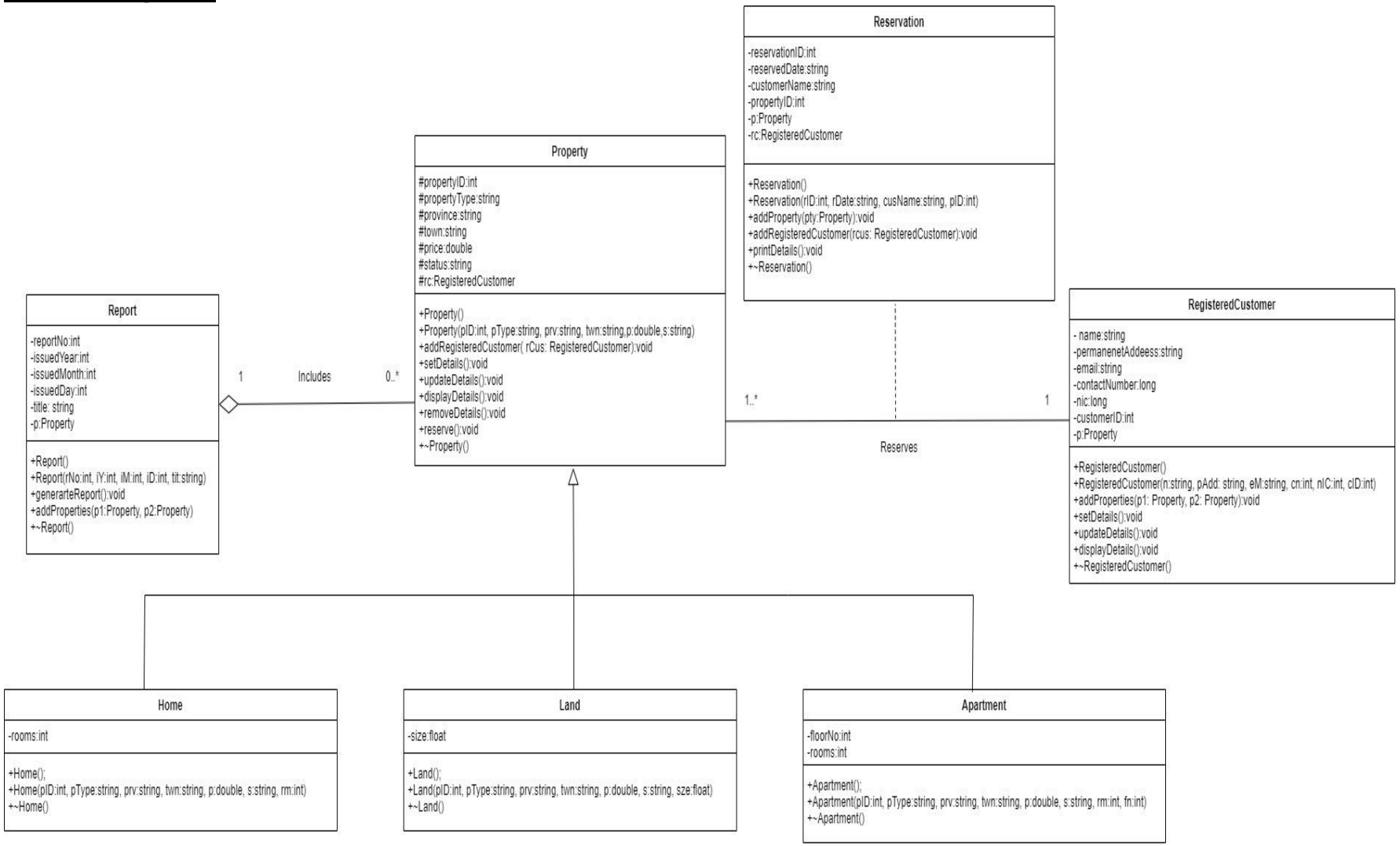
Home Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Land Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Apartment Class	
Responsibilities	**Collaborations**
Set Details	
Update Details	
Display Details	
Remove Details	

Report Class	
Responsibilities	**Collaborations**
Generate Report	Property

Class Diagram



Important:

1. **We used “draw.io” software to design class diagram**
2. **More clear and understandable class diagram is included to uploaded file.**

Implementation

File Structure

01. Main Program

- main.cpp

02. Header Files

- Property.h
- RegisteredCustomer.h
- Home.h
- Land.h
- Apartment.h
- Report.h
- Reservation.h

03. .cpp Files

- Property.cpp
- RegisteredCustomer.cpp
- Home.cpp
- Land.cpp
- Apartment.cpp
- Report.cpp
- Reservation.cpp

01. RegisteredCustomer class

RegisteredCustomer.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#define SIZE 2

class RegisteredCustomer{
        private:
                char name[100];
                char permanentAddress[100];
                char email[100];
                long contactNumber;
                long nic;
                int customerID;

                Property* p[SIZE];

        public:
           RegisteredCustomer();
                RegisteredCustomer(char n[100],char pAdd[100],char eM[100],int
cn,int nIC,int cID);
                void addProperties(Property* p1,Property* p2);
                void setDetails();
                void updateDetails();
                void displayDetails();
                ~RegisteredCustomer();
};
```

RegisteredCustomer.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"RegisteredCustomer.h"
#include"Property.h"
#define SIZE 2


RegisteredCustomer::RegisteredCustomer(){
            strcpy(name," ");
                strcpy(permanentAddress," ");
                strcpy(email," ");
                contactNumber=0;
                nic=0;
                customerID=0;
            }

RegisteredCustomer::RegisteredCustomer(char n[100],char pAdd[100],char
eM[100],int cn,int nIC,int cID){
                    strcpy(name,n);
                strcpy(permanentAddress,pAdd);
                strcpy(email,eM);
                contactNumber=cn;
                nic=nIC;
                customerID=cID;

                cout<<"\n\nCustomer ID: "<<customerID<<" Registered Customer
object created";
            }

void RegisteredCustomer::addProperties(Property* p1,Property* p2){
                    p[0]=p1;
                    p[1]=p2;
            }

void RegisteredCustomer::setDetails(){}
void RegisteredCustomer::updateDetails(){}
```

```
void RegisteredCustomer::displayDetails(){}

RegisteredCustomer::~RegisteredCustomer(){
                        cout<<"\n\nCustomer ID: "<<customerID<<" Registered
Customer object deleted";
                }
```

02. Property class

Property.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include "RegisteredCustomer.h"


class Property{
	protected:
		int propertyID;
		char propertyType[10];
		char province[20];
		char town[20];
		double price;
		char status[10];

		RegisteredCustomer* rc;

	public:
		Property();
		Property(int pID,char pType[10],char prv[20],char twn[20],double p,char s[10]);
		virtual void addRegisteredCustomer(RegisteredCustomer* rCus);
		virtual void setDetails();
		virtual void updateDetails();
```

```
        virtual void displayDetails();
        virtual void removeDetails();
        virtual void reserve();
        ~Property();
};
```

Property.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include "RegisteredCustomer.h"
#include"Property.h"

Property::Property(){
                    propertyID=0;
                    strcpy(propertyType," ");
                    strcpy(province," ");
                    strcpy(town," ");
                    price=0;
                    strcpy(status," ");
                }

Property::Property(int pID,char pType[10],char prv[20],char twn[20],double p,char
s[10]){
                    propertyID=pID;
                    strcpy(propertyType,pType);
                    strcpy(province,prv);
                    strcpy(town,twn);
                    price=p;
                    strcpy(status,s);
```

```
                cout<<"\n\nProperty ID: "<<propertyID<<"Property object
created";
        }

void Property::addRegisteredCustomer(RegisteredCustomer* rCus){
                rc=rCus;
        }

void Property::setDetails(){}
void Property::updateDetails(){}
void Property::displayDetails(){}
void Property::removeDetails(){}
void Property::reserve(){}

Property::~Property(){
                cout<<"\n\nProperty ID: "<<propertyID<<"Property object
deleted";
        }
```

03. Land class

Land.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"

class Land:public Property{
	private:
		float size;
	public:
		Land();
		Land(int pID,char pType[10],char prv[20],char twn[20],double p,char s[10],float sze);
		~Land();
};
```

Land.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"Land.h"

Land::Land(){
                    propertyID=0;
                    strcpy(propertyType," ");
                    strcpy(province," ");
                    strcpy(town," ");
                    price=0;
                    strcpy(status," ");
                    size=0;
                }

Land::Land(int pID,char pType[10],char prv[20],char twn[20],double p,char
s[10],float sze):Property(pID,pType,prv,twn,p,s){
                    size=sze;

                    cout<<"\n\nProperty ID: "<<pID<<" Land object created";
                }


Land::~Land(){
```

```
                cout<<"\n\nProperty ID: "<<propertyID<<" Land object
deleted";
        }
```

04. Home class

Home.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"

class Home:public Property{
        private:
                int rooms;
        public:
           Home();
                Home(int pID,char pType[10],char prv[20],char twn[20],double
p,char s[10],int rm);
                ~Home();
};
```

Home.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"Home.h"

Home::Home(){
		propertyID=0;
			strcpy(propertyType," ");
			strcpy(province," ");
			strcpy(town," ");
			price=0;
			strcpy(status," ");
			rooms=0;
		}

Home::Home(int pID,char pType[10],char prv[20],char twn[20],double p,char
s[10],int rm):Property(pID,pType,prv,twn,p,s){
			rooms=rm;

			cout<<"\n\nProperty ID: "<<pID<<" Home object created";
		}
Home::~Home(){
			cout<<"\n\nProperty ID: "<<propertyID<<" Home object
deleted";}
```

05. Apartment class

Apartment.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"

class Apartment:public Property{
        private:
                int floorNo;
                int rooms;
        public:
           Apartment();
                Apartment(int pID,char pType[10],char prv[20],char twn[20],double p,char s[10],int rm,int fn);

                ~Apartment();
};
```

Apartment.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"Apartment.h"



Apartment::Apartment(){
		floorNo=0;
		rooms=0;
		}

Apartment::Apartment(int pID,char pType[10],char prv[20],char twn[20],double
p,char s[10],int rm,int fn):Property(pID,pType,prv,twn,p,s){
			floorNo=fn;
			rooms=rm;


			cout<<"\n\nProperty ID: "<<pID<<" Apartment object
created";
	}

Apartment::~Apartment(){
			cout<<"\n\nProperty ID: "<<propertyID<<" Apartment object
deleted";
		}
```

06. Report class

Report.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#define SIZE 2

class Report{
        private:
                int reportNo;
                int issuedYear;
                int issuedMonth;
                int issuedDay;
            char title[30];

            Property* p[SIZE];

        public:
            Report();
                Report(int rNo,int iY,int iM,int iD,char tit[30]);
                void generarteReport();
                void addProperties(Property* p1,Property* p2);
                ~Report();
};
```

Report.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"Report.h"
#define SIZE 2

Report::Report(){
                reportNo=0;
                issuedYear=0;
                   issuedMonth=0;
                   issuedDay=0;
              strcpy(title," ");
                }

Report::Report(int rNo,int iY,int iM,int iD,char tit[30]){
                        reportNo=rNo;
                        issuedYear=iY;
                   issuedMonth=iM;
                   issuedDay=iD;
              strcpy(title,tit);

              cout<<"\n\nReport No: "<<reportNo<<"Report object created";
                }
```

```
void Report::generarteReport(){
}

void Report::addProperties(Property* p1,Property* p2){
                p[0]=p1;
                p[1]=p2;
        }

Report::~Report(){
                cout<<"\n\nReport No: "<<reportNo<<"Report object deleted";
        }
```

07. Reservation class

Reservation.h

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"RegisteredCustomer.h"


class Reservation{
        private:
                int reservationID;
                char reservedDate[10];
                char customerName[100];
                int propertyID;

                Property* p;
                RegisteredCustomer* rc;
        public:
           Reservation()
                Reservation(int rID,char rDate[10],char cusName[100],int pID);
                void addProperty(Property* pty);
                void addRegisteredCustomer(RegisteredCustomer* rcus);
                void printDetails();
                ~Reservation();
```

```
};
```

Reservation.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#include"Property.h"
#include"RegisteredCustomer.h"
#include"Reservation.h"

Reservation::Reservation(){
                reservationID=0;
                   strcpy(reservedDate," ");
                   strcpy(customerName," ");
                   propertyID=0;
                }

Reservation::Reservation(int rID,char rDate[10],char cusName[100],int pID){
                        reservationID=rID;
                   strcpy(reservedDate,rDate);
                   strcpy(customerName,cusName);
                   propertyID=pID;

                   cout<<"\n\nResevation ID: "<<reservationID<<" Reservation
object created";
                }

void Reservation::addProperty(Property* pty){
```

```
        p=pty;
    }

void Reservation::addRegisteredCustomer(RegisteredCustomer* rcus){
        rc=rcus;
    }

void Reservation::printDetails(){
}

Reservation::~Reservation(){
        cout<<"\n\nResevation ID: "<<reservationID<<" Reservation
object deleted";
    }
```

main.cpp

```
#include<iostream>
#include<cstring>
using namespace std;
#define SIZE 2
#include"Property.h"
#include"RegisteredCustomer.h"
#include"Land.h"
#include"Home.h"
#include"Apartment.h"
#include"Report.h"
#include"Reservation.h"



int main(){

        //creating two dynamic property objects 'p1' and 'p2'
        Property* p1=new
Property(1245,"Land","Western_Province","Kadawatha",123000,"Available");
        Property* p2=new
Property(3421,"Apartment","Southern_Province","Matara",2345000,"Available");


        //creating one dynamic registeredCustomer object 'regCus'
        RegisteredCustomer* regCus=new
RegisteredCustomer("Tharin_Ransika","34/5,Pansala-
Para,Ganemulla","tharinransika128@gmail.com",711758782,200002801625,513);
```

```
//calling 'addRegisteredCustomer' function in Property objects(p1 ,p2 ) and passing Customer object(regCus) as an argument

p1->addRegisteredCustomer(regCus);

p2->addRegisteredCustomer(regCus);


//calling 'addProperties' function in RegisteredCustomer object(regCus) and passing Property objects(p1,p2) as arguments

regCus->addProperties(p1,p2);


//creating one dynamic Report object 'r'

Report* r=new Report(102,2022,4,7,"Monthly_Report");


//calling 'addProperties' function in Report object(r) and passing Property objects(p1,p2) as arguments

r->addProperties(p1,p2);


//Resevation object contains details of one property reservation

//create two dynamic Reservation objects 'rsv1' and 'rsv2'


//create 1st dynamic Reservation object 'rsv1'

Reservation* rsv1=new Reservation(101,"7/04/2022","Tharin_Ransika",1245);

//calling 'addProperty' function in Reservation object(rsv1) and passing Property Object(p1) as an argument

rsv1->addProperty(p1);
```

```
        //calling 'addRegisteredCustomer' function in Reservation object(rsv1) and passing RegisteredCustomer object(regCus) as an argument
        rsv1->addRegisteredCustomer(regCus);


        //create 2nd dynamic Reservation object 'rsv2'
        Reservation* rsv2=new Reservation(105,"7/04/2022","Tharin_Ransika",3421);
        //calling 'addProperty' function in Reservation object(rsv2) and passing Property Object(p2) as an argument
        rsv2->addProperty(p2);
        //calling 'addRegisteredCustomer' function in Reservation object(rsv2) and passing RegisteredCustomer object(regCus) as an argument
        rsv2->addRegisteredCustomer(regCus);



        //create Land object 'l1'
        Land* l1=new Land(1300,"Land","Western_Province","Keleniya",223000,"Available",22);


        //create Home object 'h1'
        Home* h1=new Home(1324,"Home","Southern_Province","Gall",322000,"Available",5);


        //create Apartment object 'a1'
        Apartment* a1=new Apartment(1156,"Apartment","Central_Province","Kandy",222000,"Available",3,4);
```

```
        delete(p1);
        delete(p2);
        delete(regCus);
        delete(r);
        delete(rsv1);
        delete(rsv2);
        delete(l1);
        delete(h1);
        delete(a1);


        return 0;
}//end function main
```

Individual Contribution

Name	Student ID	Contribution
M.R. Tharin Ransika	IT21177514	1. main.cpp 2. RegisteredCustomer.h 3. RegisteredCustomer.cpp 4. Report.h 5. Report.cpp 6. Reservation.h 7. Reservation.cpp
K.V.S.D. Dasanya	IT21178504	1. Property.h 2. Property.cpp 3. Apartment.h 4. Apartment.cpp
D.H.K. Hasini Ishara	IT21179976	1. Home.h 2. Home.cpp
K.K.L. Rathnasiri	IT21178740	1. Land.h 2. Land.cpp

The End