

The program creates the "delivery_app" class, which has data members like the name of the customer, the number of items, the price and the availability. From delivery_app, another class called deliver is descended, and it includes more data members like the delivery address and distance. Deliver is the basis for the third class rating, which also includes additional data members such client feedback and satisfactory experience. The code also has a friend function called app_name_comments that has access to the rating class's app_name data member. It also provides the customer rating exception handling, where if an input other than integer is acquired then the code terminated right there. By establishing a pointer to the base class lastpart and allocating it to an object of the derived class endclass, the main() code also illustrates polymorphism. Finally, it calls the virtual function last() using the pointer to demonstrate runtime polymorphism.

Code:

```
#include <iostream>
using namespace std;
class delivery_app          // Class creation
{
// Access specifiers for inserting different data members
private:
    string customer;
    int items;
public:
    int price;
    bool availability;

//inputting the different data members
    delivery_app()          // Constructor usage
    {
        cout<<"Customer name? \n";
        cin>>customer;
        cout<<"Number of items? \n";
        cin>>items;
        cout<<"Price? \n";
        cin>>price;
        cout<<"Available? \n";
        cin>>availability;
    }
    void display()          // Display the inputted data members
    {
        cout<<"Name: "<<customer<<"\n"<<items<<" items"<<endl;
        cout<<"Rs "<<price<<endl;
        if(availability==true)
        {
            cout<<"Product is available \n";
        }
        else{
            cout<<"Product not available \n";
        }
    }
};

//Creating another class "deliver" from the above class delivery_app" using
inheritance
class deliver:public delivery_app
{
private:
    string address;
```

```

int distance;
public:
deliver()      //Constructor usage
{
    cout<<"Enter the address? \n";
    cin>>address;
    cout<<"Enter the distance to reach the destination \n";
    cin>>distance;
}
// Calling the first class function for inheriting the its respective details
void print()
{
    display();
    cout<<address<<endl;
    cout<<distance<<" kilometers"<<endl;
}
~deliver()    //Destructor usage
{
    cout<<"Can't be delivered \n";
}
protected:
int numlogins;
};
//Third class "rating" inherited from the "deliver" class, this is multilevel
inheritance
class rating:public deliver
{
    private:
    int customer_rating;
    bool satisfactory;
    protected:
    string app_name;
    public:
    void review(int l)
    {
        numlogins = l;
        cout<<"Enter the number of logins? \n";
        cin>>numlogins;

        try{
            cout<<"What is the customer rating? \n";
            cin>>customer_rating;
            if(cin.fail())
            {
                throw "Invalid Input: You can only input a numeric value for
customer rating";
            }
        }
        catch(const char* e)
        {
            cout<<e<<endl;
            exit(1);
        }

        cout<<"Was the experience satisfactory? \n";
        cin>>satisfactory;
        cout<<"App name? \n";
        cin>>app_name;
    }
}

```

```

void show()
{
    print(); // Calling the "print" function from the above class
    cout<<numlogins<<" logins"<<"\n"<<customer_rating<<" rating"<<endl;
    if(satisfactory==true)
        cout<<"Yes very satisfied!!! \n";
    else
        cout<<"not satisfied!!! \n";
}
~rating()
{
    cout<<"No activity \n";
}

//Inclusion of friend function do display the name of the app

friend void app_name_comments(rating& r);
};

void app_name_comments(rating& r)    //Declaration of friend function, which is a
friend of class "rating"
{
    cout<<"The App name is: "<<r.app_name<<"\n";
}

class lastpart
{
public:
    virtual void last() = 0;
    //{
    //    cout<<"You have to pay for being a part of Swiggy Instamart \n";
    //}
protected:
    bool instamart;
};

class endclass:public lastpart
{
public:
    void last()
    {
        if(instamart==true)
            cout<<"You cannot be a part of Swiggy Instamart \n";
        else
            cout<<"You can be a part of Swiggy Instamart \n";
    }
};

int main()
{
    int r;
    rating obj_1;
    obj_1.review(r);
    obj_1.show();
    app_name_comments(obj_1);
    //lastpart l;
    //endclass e;
    //e.last();
}

```

```
lastpart *ptr=new endclass();  
ptr->last();  
return 0;  
}
```