

```

#include<iostream>
using namespace std;

class printer {
private:
    string model, manufacturer;
    int printedCount, availableCount;
protected:
    printer(string model, string manufacturer, int availableCount) {
        this->model = model;
        this->manufacturer = manufacturer;
        this->availableCount = availableCount;
    }
    bool print(int pages) {
        if (availableCount < pages) {
            cout << "용지가 부족하여 프린트 할 수 없습니다." << endl;
            return false;
        }
        availableCount -= pages;
        printedCount += pages;
        return true;
    }
    string getModel() { return model; }
    string getManufacturer() { return manufacturer; }
    int getAvailablecount() { return availableCount; }
};

```

printer클래스를 만든다음 model, manufacturer, printedCount, availableCount 멤버 변수를 만들어 주었다. 각각 모델명, 제조사, 프린트 횟수, 프린트가 가능한지(남은 용지의 수)를 나타낸다. 그다음 protected로 생성자를 선언하여 하위클래스에서 상속받게 해주었다. printer생성자는 객체가 생성되었을때 받은 값으로 객체의 멤버변수를 초기화 해준다. bool print는 얼마나 프린트 할지를 나타내는 pages를 인수로 받아 만약 pages가 availableCount 보다 많으면 용지가 부족하다고 출력해준 다음 false를 리턴한다. 아니면 프린트한 만큼 availableCount를 차감하고 printedCount를 늘린다음 true를 리턴한다

밑은 각 멤버를 받을 수 있는 함수들이다.

```

class InkJet : public printer {
    int availableInk;
public:
    InkJet(string model, string manufacturer, int availableCount, int availableInk) : printer(model, manufacturer, availableCount) {
        this->availableInk = availableInk;
    }

    bool printInkJet(int pages) {
        if (print(pages)) {
            else return false;
            if (availableInk < pages) {
                cout << "잉크가 부족하여 프린트 할 수 없습니다." << endl;
                return false;
            }
            availableInk -= pages;
            return true;
        }
    }

    void show() {
        cout << getModel() << " " << getManufacturer() << " 남은 종이 : " << getAvailableCount() << "잔, 남은 잉크 : " << availableInk << endl;
    }
}

```

```

class Laser : public printer {
    int availableToner;
public:
    Laser(string model, string manufacturer, int availableCount, int availableToner) : printer(model, manufacturer, availableCount) {
        this->availableToner = availableToner;
    }

    bool printLaser(int pages) {
        if (print(pages)) {
            else return false;
            if (availableToner < pages) {
                cout << "토너가 부족하여 프린트 할 수 없습니다." << endl;
                return false;
            }
            availableToner -= pages;
            return true;
        }
    }

    void show() {
        cout << getModel() << " " << getManufacturer() << " 남은 종이 : " << getAvailableCount() << "잔, 남은 토너 : " << availableToner << endl;
    }
}

```

InkJet라는 클래스를 만들고 printer를 상속시켰다. InkJet의 멤버변수는 printer에서 받아오고 새로이 availableInk 라는 남은 잉크를 나타내는 변수를 새로 가진다. printInkJet는 프린트할 갯수 pages를 인수로 받아서 printer 클래스에있는 print()함수로 용지의 갯수를 확인하고 용지의 갯수가 충분하면 availableInk와 비교하여 잉크의 갯수가 충분한지 확인한다음 잉크의 수를 프린트한 만큼 줄인다. show()는 현재 InkJet프린터의 상태를 출력한다. Laser클래스도 추가되는 변수가 availableToner 일뿐 InkJet클래스와 동일하다.

```

#include <iostream>
#include <string>
using namespace std;

int main() {
    int printer, pages;
    char yn;
    InkJet inkjet = new InkJet("Officejet V40", "hp", 5, 10);
    Laser laser = new Laser("SL-B645", "삼성전자", 3, 20);

    cout << "현재 작동중인 2대의 프린터는 아래와 같다." << endl;
    cout << "인크젯 : ";
    inkjet->show();
    cout << "레이저 : ";
    laser->show();
    cout << endl;

    while (true) {
        while (true) {
            cout << "프린터(1:인크젯, 2:레이저)와 페이지 수를 입력>>";
            cin << "printer " << pages;

            if (printer == 1) {
                if (inkjet->print(inkjet, pages)) {
                    cout << "프린트 되었습니다." << endl;
                    break;
                }
            }
            else if (printer == 2) {
                if (laser->print(laser, pages)) {
                    cout << "프린트 되었습니다." << endl;
                    break;
                }
            }
            else {
                cout << "프린터를 잘못 선택하셨습니다. 1 또는 2를 입력해 주세요." << endl;
            }
        }

        inkjet->show();
        laser->show();
        while (true) {
            cout << "계속 프린트 하시겠습니까?(y/n)>>";
            cin << yn;

            if (yn == 'n') {
                delete inkjet;
                delete laser;
                return 0;
            }
            else if (yn == 'y') {
                break;
            }
            else {
                cout << "잘못 입력하셨습니다. y 또는 n을 입력해 주세요." << endl;
            }
        }
    }
}

```

프린터를 구별할 변수 printer, pages 프린트를 계속할지를 구별할 변수 yn 그리고 클래스 InkJet, Laser의 객체를 동적 생성해준다.

현재 작동중인 프린터 2대를 출력해준 다음, while문으로 프린터를 사용하는 것을 표현하였다. cin으로 printer와 pages를 받은다음 printer가 1이면 InkJet프린터 2이면 Laser프린터 pages는 프린트할 갯수를 나타낸다. 다른것을 입력하면 다시 입력하라고 한다.

그 다음 프린터의 상태를 출력한다.

그리고 cin으로 프린트를 계속 할꺼지 받는다. n을 입력했으면 동적할당한 inkjet, laser를 delete하고 프로그램을 종료한다. y를 입력했으면, break하고 다시 처음부터 진행한다. 잘못 입력했으면 다시 입력하라고 한다음 다시 물어본다.

현재 작동 중인 2 대의 프린터는 아래와 같다  
잉크젯 : Officejet V40 ,Hp ,남은 종이 5장 ,남은 잉크 10  
레이저 : SCX-6x45 ,삼성전자 ,남은 종이 3장 ,남은 토너 20

프린터(1:잉크젯, 2:레이저)와 매수 입력>>1 10  
용지가 부족하여 프린트 할 수 없습니다.  
Officejet V40 ,Hp ,남은 종이 5장 ,남은 잉크 10  
SCX-6x45 ,삼성전자 ,남은 종이 3장 ,남은 토너 20  
계속 프린트 하시겠습니까(y/n)>>y

프린터(1:잉크젯, 2:레이저)와 매수 입력>> 1 3  
프린트 하였습니다.  
Officejet V40 ,Hp ,남은 종이 2장 ,남은 잉크 7  
SCX-6x45 ,삼성전자 ,남은 종이 3장 ,남은 토너 20  
계속 프린트 하시겠습니까(y/n)>>y

프린터(1:잉크젯, 2:레이저)와 매수 입력>>2 3  
프린트 하였습니다.  
Officejet V40 ,Hp ,남은 종이 2장 ,남은 잉크 7  
SCX-6x45 ,삼성전자 ,남은 종이 0장 ,남은 토너 17  
계속 프린트 하시겠습니까(y/n)>>

실행결과