

객체지향프로그래밍

3-2

1. [Virtual Function] 아래의 main함수가 결과화면과 같이 동작하게 class Equation과 class TwoPoint를 구현하시오. class Equation은 TwoPoint를 상속받으며 두 class에는 print와 생성자 외의 함수는 없다. TwoPoint의 print함수는 생성자에서 저장한 두 점의 좌표를 출력하며(결과화면 1,2번째 줄) Equation의 print함수는 두 점을 잇는 직선의 방정식을 구해 출력한다(결과화면 3번째 줄). 과제 제출시 main함수를 포함해서 제출

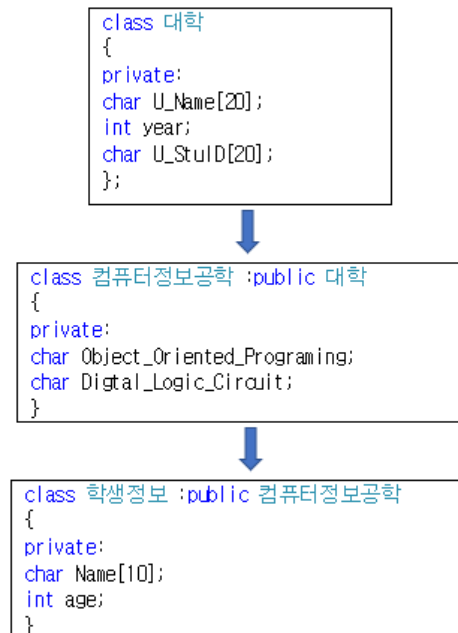
```
int main()
{
    class TwoPoint *A = new TwoPoint(2, 4, 7, 8);
    class TwoPoint *B = new Equation(2, 4, 7, 8);
    A->print();
    B->print();
    delete A;
    delete B;
}
```

<main함수>

```
Point 1 : 2,4
Point 2 : 7,8
Equation : y = 0.8x + 2.4
```

<결과화면>

2. [Inheritance]Table1의 멤버변수들을 가지는 Figure 1의 3개의 Class를 구현하고 Table 2의 3개의 command를 가지는 학생정보관리 프로그램을 구현하시오. 멤버함수는 자유롭게 구현할 수 있지만 생성자를 통해 멤버변수들의 값을 저장해야하며 반드시 Class의 멤버변수들을 이용해야한다.



<Figure 1>

멤버함수	저장 정보
U_Name	대 학 명
Year	학 년

U_StuID	학번
Object_Oriented_Programing	OOP 의 성적
Digital_Logic_Circuit	DLC 성적
Name	이름
age	나이

<Table 1>

Command	사용법	설명
INSERT	INSERT	새로운 학생 정보를 입력 정보는 결과화면과 같이 학교, 학번, 학년, OOP 학점, DLC 학점, 이름, 나이를 입력받아 저장한다. 최대 100 개의 정보를 저장하며 그 이상의 정보가 들어올 경우 무시한다.
FIND	FIND 이름 학번	저장된 정보 중 이름과 학번이 맞는 정보가 있는지 검색하고 맞는 정보가 있을 경우 학교,학번,학년,OOP 학점, DLC 학점 순으로 출력한다
EXIT	EXIT	Program 종료

<Table 2>

학생정보는 학생정보 class 에 저장되어 있으며. 학생정보** type 으로 학생정보 class 를 관리한다

```

INSERT
학교 :광운대학교
학번 :2021110606
학년 :1
객체지향프로그래밍 학점 :A
디지털논리회로 학점 :A
이름 :박우혁
나이 :00
INSERT
학교 :광운대학교
학번 :2015722030
학년 :4
객체지향프로그래밍 학점 :F
디지털논리회로 학점 :F
이름 :홍길동
나이 :00

```

```

FIND 박우혁 2021110606
학교:광운대학교
학번:2021110606
학년:1
객체지향프로그래밍 학점:A
디지털논리회로 학점: A
FIND 홍길동 2015722030
학교:광운대학교
학번:2015722030
학년:4
객체지향프로그래밍 학점:F
디지털논리회로 학점: F

```

EXIT

C:\Users\U Hyeok Bak\source\repos\ass1
이 창을 닫으려면 아무 키나 누르세요...

<결과화면>

3. [Inheritance]물품 판매 시뮬레이터를 구현하시오. 프로그램은 Figure 1과 같이 구현된 item class를 상속받는 Pen, Pencil, Eraser Class를 이용해 물품의 값, 재고, 할인율을 관리한다. 프로그램의 Command는 Table 1과 같으며 프로그램은 Command를 입력 받아 동작할 때 마다 매출을 출력한다. Pen, Pencil, Eraser Class는 멤버변수를 가지고 있지 않고 item의 멤버변수만 사용하며 item은 Figure 1외에 추가하지 않는다. 프로그램 시작 시 모든 물품의 재고는 100개로 가정한다.

```
class item
{
private:
    int price;
    int stock;
    int discount;
public:
    item(int price, int stock) { this->price = price; this->stock = stock; discount = 0; }
    int getprice() { return price; }
    int getstock() { return stock; }
    int getdiscount() { return discount; }
    void addstock(int num) { this->stock += num; }
    void sell_item(int num) { this->stock -= num; }
    void setdiscount(int num) { discount = num; }
};
```

<Figure 1>

Command	사용법	설명
Sell	0 물품명 개수	<p>물품명에 맞는 물품을 개수*판매단위 만큼 판매하며 매출을 물품의 값 * 판매단위 * 개수*할인율만큼 더하고 재고를 개수*판매단위 만큼 뺀다.</p> <p>Pen 판매단위 : 3 개 가격 : 개당 200</p> <p>Pencil 판매단위 : 12 개 가격 : 개당 100</p> <p>Eraser 판매단위 : 1 개 가격 : 개당 500</p> <p>만약 개수*판매단위가 재고보다 많을 경우 명령을 무시한다.</p>
AddStock	1 물품명 개수	물품명에 맞는 물품의 재고를 개수만큼 더한다.

Discount	2 물품명 할인율	물품명에 맞는 물품의 할인율을 입력 받은 값으로 바꾼다
Print	3	모든 물품의 가격, 재고, 할인율을 출력
Quit	4	프로그램 종료

```

=====
-----Pen-----
Price : 600
Stock : 100
Discount : 0%
-----Pencil-----
Price : 1200
Stock : 100
Discount : 0%
-----Eraser-----
Price : 500
Stock : 100
Discount : 0%
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :0 Pen 3
=====
Sales : 1800
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :0 Pencil 7
=====
Sales : 10200
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :0 Eraser 3
=====
Sales : 11700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :3
=====
-----Pen-----
Price : 600
Stock : 91
Discount : 0%
-----Pencil-----
Price : 1200
Stock : 16
Discount : 0%
-----Eraser-----
Price : 500
Stock : 97
Discount : 0%
=====
=====
Sales : 11700
=====

```

<결과화면 1 Sell>

```

Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :1 Pen 19
=====
Sales : 11700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :1 Pen 60
=====
Sales : 11700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :1 Pencil 60
=====
Sales : 11700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :3
=====
-----Pen-----
Price : 600
Stock : 170
Discount : 0%
-----Pencil-----
Price : 1200
Stock : 76
Discount : 0%
-----Eraser-----
Price : 500
Stock : 97
Discount : 0%
=====
Sales : 11700
=====

```

<결과화면 2 Add Stock>

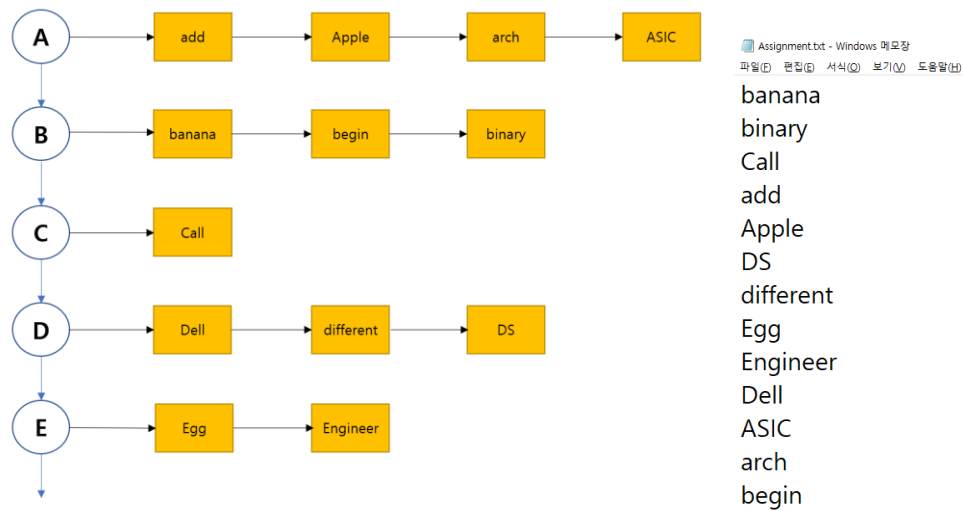
```

Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :2 Pen 50
=====
Sales : 11700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :0 Pen 10
=====
Sales : 14700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :3
=====
-----Pen-----
Price : 600
Stock : 140
Discount : 50%
-----Pencil-----
Price : 1200
Stock : 76
Discount : 0%
-----Eraser-----
Price : 500
Stock : 97
Discount : 0%
=====
Sales : 14700
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit) :4
=====

```

<결과화면 3 Discount>

4. [2D Linked List] 2D Linked List를 이용한 단어장 프로그램을 구현하시오. 프로그램은 Assignment.txt로부터 단어를 읽어 Figure 1과 같이 알파벳순으로 1차원 Linked List를 구현하고 단어의 시작 알파벳과 같은 알파벳의 Node에 사전식배열로 정렬되어 Linked List로 저장된다. 프로그램은 시작 시 Assignment.txt로부터 단어를 읽어서 정렬한 후 결과화면과 같이 저장된 단어가 존재하는 알파벳과 그 단어들을 출력한다.
(프로그램 구현 시 파일명 Assignment.txt 준수)



<Figure 1>

```

A -> add -> Apple -> arch -> ASIC
B -> banana -> begin -> binary
C -> Call
D -> Dell -> different -> DS
E -> Egg -> Engineer
  
```

<결과화면>