Object Oriented Programming (Week 10)

2023

KWANGWOON UNIVERSITY

DEPT. OF COMPUTER ENGINEERING

IPSL

Contents

- Assignment 3-2. 1

- Assignment 3-2. 2

- Assignment 3-2. 3

- Assignment 3-2. 4

IPSL

ASSIGNMENT 3-2. 1

IPSL

Assignment 3-2. 1

- (Linked List) Write Binary_Search(Node *p, int n) function that has two arguments: a pointer to the first nodes in a liked list and a target value. The function **finds the target node that has the target value in the list by using binary search algorithm and returns the target node.** To implement the list, the program should generate 16 integers randomly between 0 and 200, and all the integers should be unique. If the target node does not exist in the list, prints a fail message and returns the closest node which minimizes the value of absolute difference between the target value and the node's value in the list. If the list is unsorted before finding the target node, the function should sort the list in a descending order by using the insertion sort algorithm and print the result. Note that the function should print each step in the search algorithm on a screen.

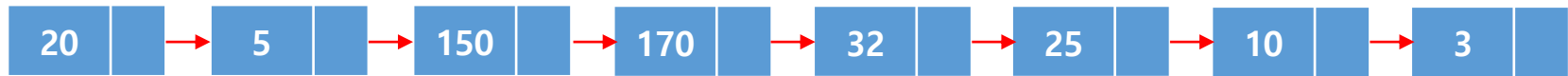# Assignment 3-2. 1

- **Binary_Search**
  - **1. mid point 값과 찾고자 하는 값(find) 비교**
  - **2. if (mid point < find) -> 오른쪽 이동 (low point = mid point +1, high point 유지)**
    else if (mid point > find) -> 왼쪽 이동 (low point 유지, high point = mid point -1)
    else (mid point == find) -> search 성공 (해당 node return)
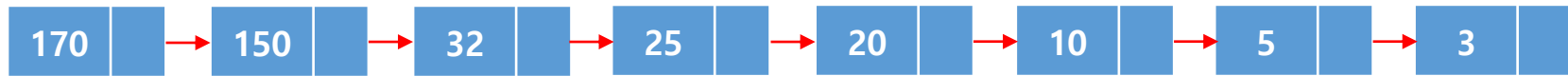 - **3. 찾을 때 반복**
  - **4. 더 이상 찾을 범위가 없거나 high < low되면 search 실패**

※ **mid point = (L+H) / 2**

Assignment 3-2. 1

예시 동작 : **Write Binary_Search(p, 21)**

| 20 | | → | 5 | | → | 150 | | → | 170 | | → | 32 | | → | 25 | | → | 10 | | → | 3 | |

↓ insertion sort(내림차순)

| 170 | | → | 150 | | → | 32 | | → | 25 | | → | 20 | | → | 10 | | → | 5 | | → | 3 | |

↓ find 21(Binary search)

print fail message and return below node

| 20 | |

※**실제 구현시에는 16개의 node에 대해 수행**

ASSIGNMENT 3-2. 2

IPSL

Assignment 3-2. 2

▪ (Class) Write a myMusic class which can save music information. MyMusic class has private member variables to store music title, singer name, album title, track number, and release year. Implement get/set functions for retrieve/modify member variables of myMusic class object. Write a program to test your code

```cpp
class myMusic
{
  private:
    char m_title[32];
    char m_singer[32];
    char m_album[32];
    int m_track_no;
    int m_year;

  public:
    myMusic();
    ~myMusic();

    void setTitle(char* title);
    void setSinger(char* singer);         Modify member variable function
    void setAlbum(char* album);
    void setYear(int year);
    void setTrackNo(int track_no);

    char* getTitle();
    char* getSinger();
    char* getAlbum();                      retrieve member variable function
    int getYear();
    int getTrackNo();
};
```
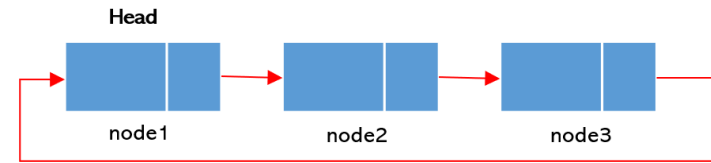
ASSIGNMENT 3-2. 3
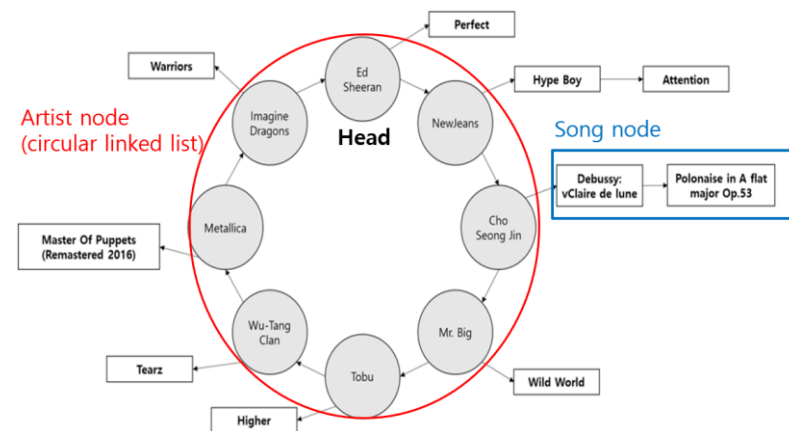
Assignment 3-2. 3

Circular linked list

Head

node1   node2   node3

▪ (Circular linked list) Implement a music management system with a 2D circular li nked list. To implement this program, you need to define three classes; MyMusic ManagementList, ArtistNode and SongNode classes. MyMusicManagementList r eads and analyzes a text file named "Music_info.txt". ArtistNode represents the ci rcular list in artist names. By reading the input file, a new artist is found; then, a n ew node is created for the artist. SongNode represents the sub-list node to store lis t of songs for an artist. All the songs must be sorted by alphabetical order. The foll owing table and figure show an example of this program. Write a program that cre ates the 2D circular linked list with MyMusicManagementList and demonstrates y our code properly.

```
Artist              /       Songs
Ed Sheeran      /       Perfect
NewJeans        /       Hype Boy
NewJeans        /       Attention
Cho Seong Jin   /       Debussy: vClaire de lune
Cho Seong Jin   /       Polonaise in A flat major Op.53
Mr. Big         /       Wild World
Tobu       /       Higher
Wu-Tang Clan    /       Tearz
Metallica       /       Master Of Puppets (Remastered 2016)
Imagine Dragons /       Warriors
...
```

Music_info.txt

Artist node (circular linked list)

Song node

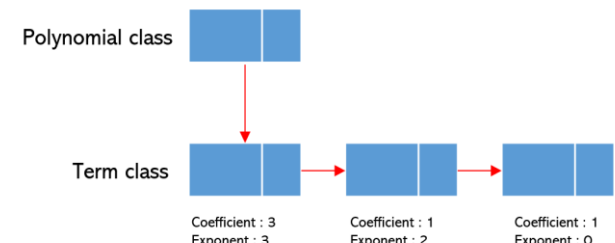※구현 시 song은 값을 소문자로 변환 후 alphabetical order에 따라 저장

A**SSIGNMENT** 3-2. 4

Assignment 3-2. 4

▪ (Class, Linked-List) Write a program that adds and subtracts polynomials. Each polynomial is represented as linked list. The first node in the list represents the first term in the polynomial, the second node represents the second term and so on. Each node contains three fields. The first field is the term's coefficient, the second field is the term's exponent and the third field is a pointer to the next term. The Add and Sub (member functions of Polynomial class) takes the object of Polynomial class as a parameter and performs polynomial arithmetic operations (addition and subtraction). The result of the operation should be stored in descending order. Suppose that a user never inserts the term with the same exponent. Class prototypes of two classes are described below.
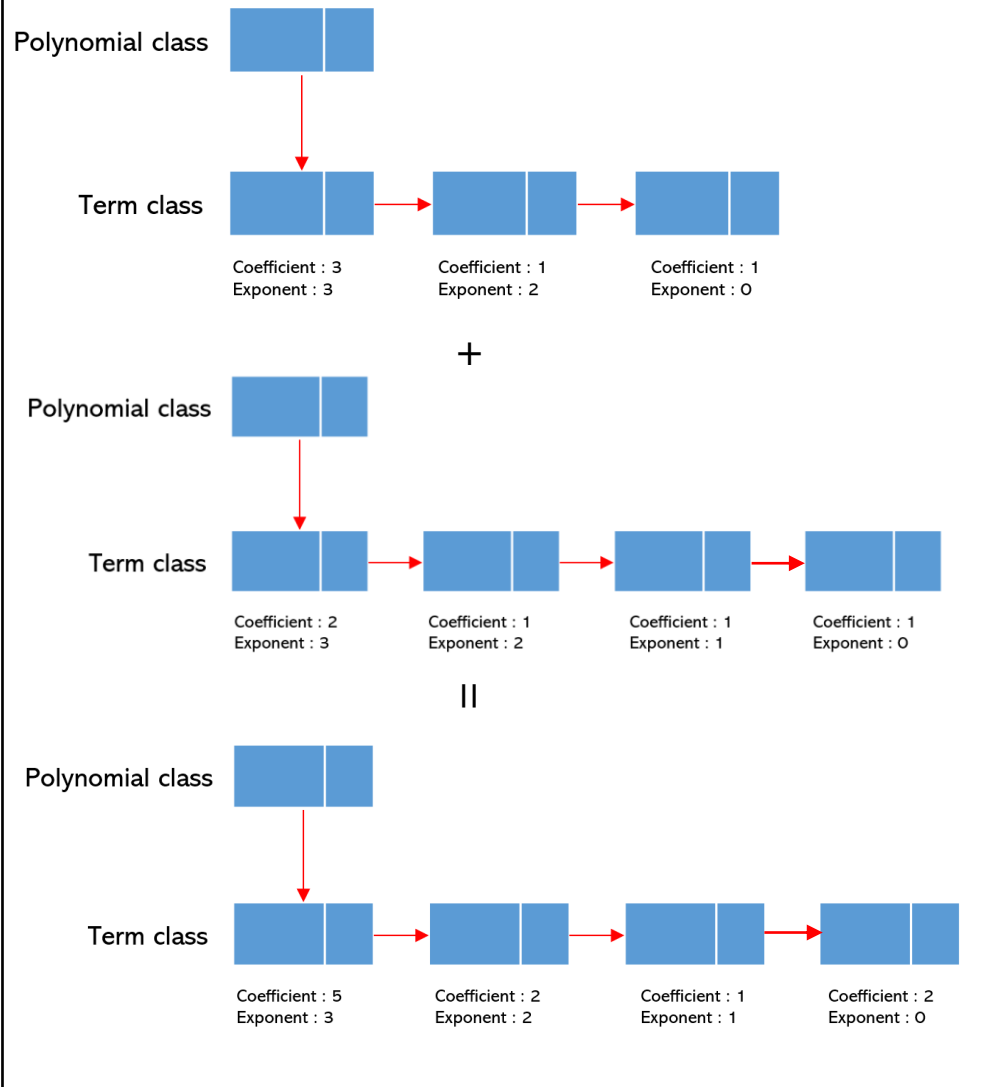
```
class Polynomial
{
  private:
    Term* m_pHead;

  public:
    Polynomial();
    ~Polynomial();

    void Insert(Term* pTerm);
    void PrintList();
    void Add(Polynomial& poly);
    void Sub(Polynomial& poly);
};
```

```
class Term
{
  private:
    int m_Coefficient;
    int m_Exponent;
    Term* m_pNext;

  public:
    Term();
    ~Term();

    void SetCoeff(int coeff);
    void SetExponent(int exponent);
    int GetCoeff();
    int GetExponent();
    void SetNext(Term* pNext);
    Term* GetNext();
};
```

Ex) $3x^3 + x^2 + 1$

Polynomial class

Term class

| Coefficient : 3 | Coefficient : 1 | Coefficient : 1 |
| Exponent : 3 | Exponent : 2 | Exponent : 0 |

Assignment 3-2. 4

ADD operation

$$3x^3 + x^2 + 1$$

$$+$$

$$2x^3 + x^2 + x + 1$$

$$=$$

$$5x^3 + 2x^2 + x + 2$$

Assignment 3-2. 4

Sub operation

Polynomial class

Term class

| Coefficient : 3 | Coefficient : 1 | Coefficient : 1 |
| Exponent : 3 | Exponent : 2 | Exponent : 0 |

$$3x^3 + x^2 + 1$$

$-$

Polynomial class

Term class

| Coefficient : 2 | Coefficient : 1 | Coefficient : 1 | Coefficient : 1 |
| Exponent : 3 | Exponent : 2 | Exponent : 1 | Exponent : 0 |

$$2x^3 + x^2 + x + 1$$

$=$

Polynomial class

Term class

| Coefficient : 1 | Coefficient : -1 |
| Exponent : 3 | Exponent : 1 |

$$x^3 - x$$
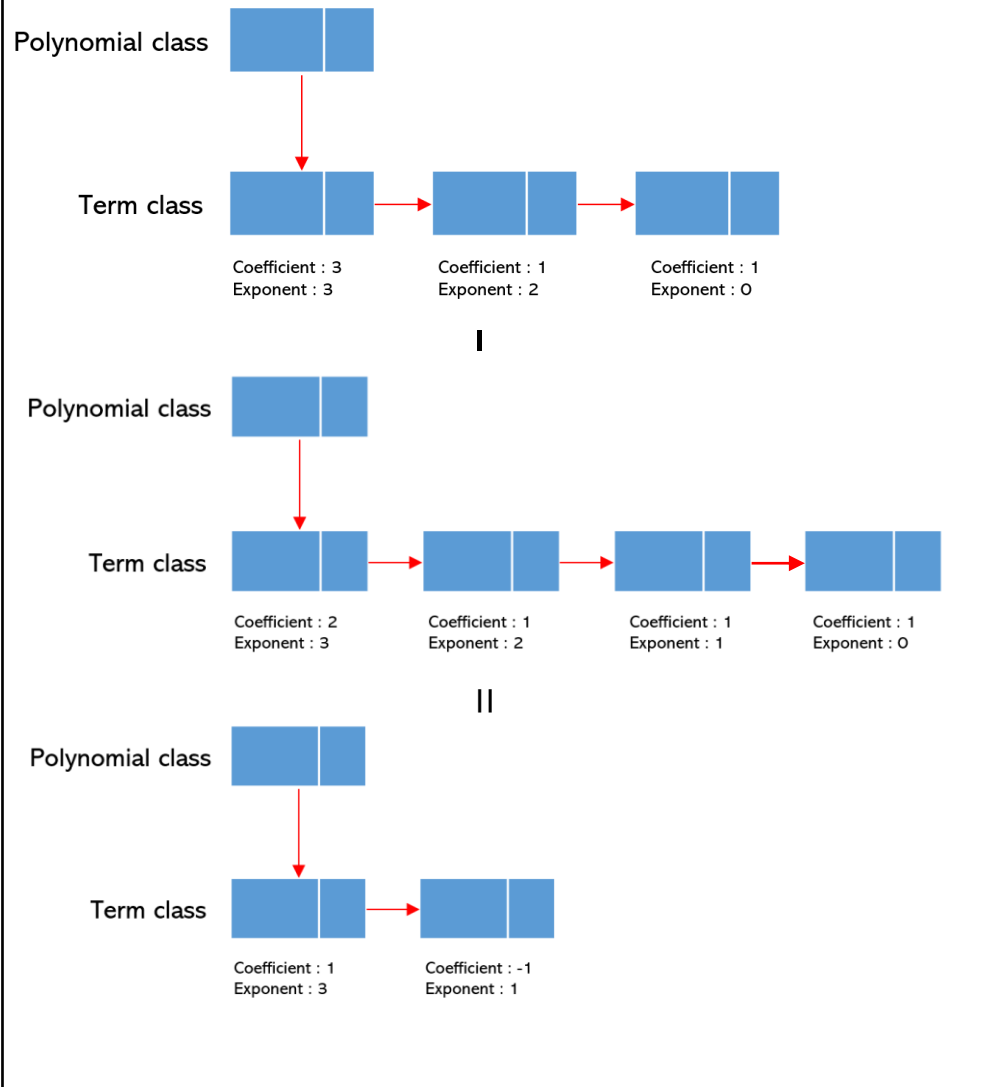
IPSL

과제 제출 방법

IPSL

과제 제출 방법

- **FTP Upload (Klas 과제 제출 X)**
  - Address : ftp://223.194.8.1:1321
  - username : IPSL_OBJ
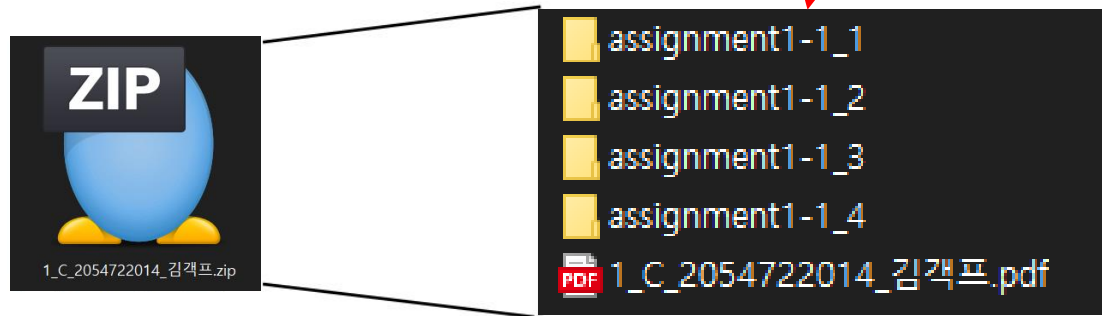 - password : ipslobj_2023

- **Due date**
  - Soft copy: 마감일 5/12(금) 23:59:59까지 제출 (서버시간 기준)
  - Delay
    - 마감일 이후 +7일까지 제출 가능
    - 단, 1일 초과마다 과제 총점의 10%씩 감점

IPSL

과제 제출 방법

- Soft copy
  - 과제(보고서, 소스 코드)를 압축한 파일 제출
    - 설계반_실습반_학번_이름.zip
      - 예) 설계1반 수강, 실습 A반: 1_A_학번_이름.zip
      - 예) 설계 수강, 실습 미수강: 2_0_학번_이름.zip
      - 예) 설계 미 수강, 실습 C반: 0_C_학번_이름.zip

  - 과제 수정하여 업로드 시 버전 명시
    - 설계반_실습반_학번_이름_verX.zip

과제 제출 방법

- Soft copy
  - 과제 보고서
    - 영문 또는 한글로 작성
    - **반드시 PDF**로 제출 (PDF 외 파일 형식으로 제출시 0점 처리)
    - 보고서 양식
      - 문제 및 설명(문제 capture 금지) / 결과 화면 / 고찰
      - 보고서 양식은 아래 경로에서 참고
        - https://www.ipsl.kw.ac.kr/post/1%EC%B0%A8-%EA%B3%BC%EC%A0%9C
    - **소스코드 제외**
    - 분량 제한 없음
    - **표절 적발 시 0점 처리**
  - 소스 코드
    - Visual Studio 2022 community 사용 필수
      - https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio?view=vs-2022
    - STL (Standard Template Library) 사용 금지 (vector, map, algorithm 등)
    - Debug 폴더를 제외한 모든 파일 제출
      - .sln 파일 포함(.cpp 만 제출하지 말것)
    - **각 문제마다 프로젝트 파일 생성 필수**
    - **주석 반드시 달기**
    - **소스코드 표절 적발 시 0점 처리**

END OF PRESENTATION

Q&A

IPSL