1. **경희학원의 인력관리 (Human Resource Management) 프로그램을 의뢰 받았다. 다음 클래스를 디자인 하라.**
   * 최상위 클래스 “Member”
     1. String형의 name, int형의 ID, String이 들어가는 STL vector형의 lectureList라는 변수 추가
     2. participateLecture(String)와 print 라는 순수가상함수를 가지는 Abstract class
   * “Member”를 상속받은 “LecturerMember”
     1. int형의 salary, int형의 workingYear 라는 멤버변수
     2. participateLecture는 파라미터로 과목이름을 받아 lectureList에 추가하는 함수
     3. Salary는 기본급 500000원 + (workingYear\*100000) 에 lecture가 하나씩 추가될 때 마다 200000씩 올려 나가야 함.
     4. print함수도 아래 출력결과와 같이 적절히 구성
   * “Member”를 상속받은 “StudentMember”
   * int형의 tuition을 멤버변수로 가짐
   * participateLecture는 파라미터로 과목이름을 받아 lectureList에 추가하고 등록금tuition은 과목이 추가될 때 마다 100000원씩 올라감.
   * print함수도 적절히 구성하라.
   * 각 클래스의 생성자들은 적절히 만들어라.
   * Main은 다음과 같이 구성

Member\* m[3];

int in;

for(int i = 0;i<3;i++) {

cout << “사용자 추가. 둘 중 선택하시오, 강사:1, 학생:2”;

cin >> in;

//[코드추가] in에 따라 m[i]가 다르게 바인딩 되는 dynamic binding.

//클래스의 내용(이름, ID)은 생성자를 통해 적절히 입력

}

//[코드추가] 각 m에 수업 2~4개씩 적절히 추가

//[코드추가] 모든 m에 대한 정보를 print함수로 화면에 출력

* + 위의 언급한 변수와 메쏘드들 외에 필요한 다른 생성자, 메쏘드, 변수 추가 가능. Access specifier들을 적절히 사용하여 data hiding을 하라.
  + 출력 포멧은 다음과 같이.

Student Gildong Hong  
ID: 20140001  
Lectures taking: CSE101, CSE302

Tuition: 200000

…

Lecturer John Doe  
ID: 049872  
Lectures giving: CSE101, CSE232

Working Years: 2  
Salary: 1100000

1. **1번에서는 수업을 String으로만 표시하여 수업에 관한 정보를 관리할 수가 없었다. 수업정보를 관리할 수 있는 Lecture클래스를 만들어라**
   * “Lecture” 클래스
     1. String형의 ID, String형의 name, Lecture\*를 저장할 수 있는 STL vector형인 prerequisiteLectureList(선수과목), int형 fromWhen, int형 toWhen (시작시간 끝시간) 변수들을 추가
     2. 위 변수들을 private 으로 하고 적절한 public 메쏘드들과 생성자들을 만들어 access할 수 있게 하라.
   * Member 클래스를 적절히 수정하여 lectureList에 String이 아닌 Lecture\*가 들어가게 하라 (participateLecture의 파라미터가 Lecture\*형이 됨). Member의 print 등의 함수도 적절히 수정
   * main에서 다음 lecture들을 만들어라.

CS101, Introduction to CS, 선수과목 없음, 1~4시까지  
CS102, Programming Basics, 선수과목 없음, 2~5시 까지  
CS201, Object-Oriented Programming, 선수과목 CS101, CS102, 10~12시 까지  
CS301, Data Structure, 선수과목 CS201, 9~11시 까지  
CS401, Operating System, 선수과목 CS301, 6~8시 까지

* + 그리고 CS101, CS102를 가르치는 Lecturer한명과 위 과목 중 3과목씩을 듣는 학생 3명을 만들어라.
  + 그리고 각각의 Member 정보를 출력하라. 정보 출력할 때 각 Member가 듣는 혹은 가르치는 수업에 대한 정보도 모두 출력하라.

1. **2번에서 학생이 듣는 과목의 시간이 겹치는 경우와 특정 과목과 그 과목의 선수과목을 한꺼번에 듣는 경우가 있다. C++의 Exception Handling 으로 이를 적절히 처리하라.**
   * Member클래스의 participateLecture함수에서 Lecture를 입력할 때 시간을 check하여 겹치면 원래 있는 Lecture중 겹치는 Lecture오브젝트를 throw함.
   * 첫 번째 catch에서 이것을 받아 화면에 겹치는 시간을 출력하고 에러메시지 출력하고 lectureList에 입력 안되게 하고 함수 끝냄. 예를 들어, CS101이 이미 있을 때 CS102를 넣으면 2~4시까지 겹친다는 메시지 출력하고 함수 끝냄.
   * 또한, participateLecture함수에서 lectureList에 있는 과목들의 선수과목을 새롭게 입력하려고 하거나 입력하려고 하는 Lecture의 선수과목이 이미 lectureList에 있다면 겹치는 과목 명(String)을 throw
   * 두 번째 catch에서 String을 받아 이를 출력하고 lectureList업데이트 하지 않고 함수 끝냄.
   * Main에서는 적절하게 이를 test할 수 있는 code와 이에 대한 comment 를 달아놓아야 함.
2. **두 번째 시험에서 3x3 Matrix를 표현하는 클래스인 Matrix3 라는 클래스를 만든 적이 있다. 이 클래스는 float형 밖에 데이터를 가질 수 없었는데 template를 사용하여 다양한 데이터의 백터를 만들수 있게 하라.** 
   * 원래 Matrix3 클래스는 다음과 같은 데이터 멤버를 가진다.

class Matrix3 {  
 float x[3][3];  
 …  
}

* + float 부분을 template class 화 하여 다른 형도 들어갈 수 있게 하면 됨.
  + 그리고 다음 operator들을 overload하라.
    1. m1+m2: 두 Matrix간의 덧셈. 결과는 Matrix3 타입으로 리턴
    2. m1-m2: 두 Matrix간의 뺄셈. 결과는 Matrix3 타입으로 리턴
    3. m1=m2: Assignment operator
  + 또한 Matrix를 출력해 주는 print()함수를 만들어라
  + test를 위해 Fraction 클래스 또한 구현하고 (책의 545~556쪽에 코드 있음)하라.
  + Main에서 Fraction형의 Matrix3를 만들고 다음과 같이 test하라. 참고로 Matrix3에서 Fraction형에 대한 operator overloading 이 가능 하려면 Fraction클래스에서 해당 operator들이 overloading 되어 있거나 Matrix3의 operator overloading 함수들을 specialized member function으로 만들어야 함.

int main() {

//[코드추가] Fraction형의 Matrix3인 m1 만들고 아무 값으로 초기화.

//[코드추가] Fraction형의 Matrix3인 m2 만들고 아무 값으로 초기화.

//[코드추가] Fraction형의 Matrix3인 m3, m4 만듬.

m3 = m1+m2;

m4 = m2-m1;

cout << “m1: “;

m1.print();

cout << “m2: “;

m2.print();

cout << “m3: “;

m3.print();

cout << “m4: “;

m4.print();

return 0;

}

1. **4번의 Matrix3를 다시 불러와 이번에는 int형 Matrix3를 3개(m1~m3)를 만든다. 임의의 숫자로 내용물을 채우고 이를 화면에 출력한다. 그리고 이 오브젝트를 binary IO로 파일 출력하라. 출력한 다음 다시 그 파일을 binary IO로 입력 받아 화면에 출력하라. Main은 다음과 같은 과정을 거치게 구성하라.**

//[코드추가] binary 출력 file open  
//[코드추가] Matrix3 형 m1, m2, m3 만들어 임의의 수로 채우고 화면에 출력  
//[코드추가] binary IO로 m1~3 파일로 출력한 다음 파일 close  
//[코드추가] Matrix3형 m4, m5, m6 만들기  
//[코드추가] binary 입력 file open  
//[코드추가] 파일로 3개의 Matrix3형 읽어 m4~6에 저장  
//[코드추가] m4~6 화면에 출력