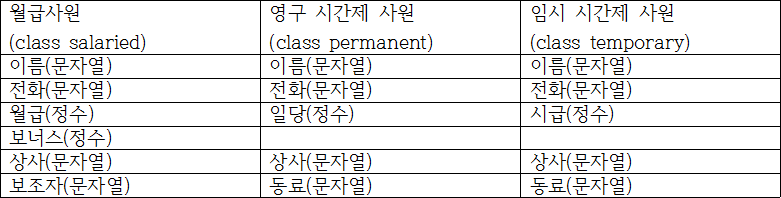
**2015, 봄 학기, 고급 객체지향 프로그래밍 시험 4**

**\* 각 문제의 main이 들어있는 cpp 파일 첫 부분에 comment로 자신이 어떤 세부문제(A, B, C…)까지 풀었는지 표시함. 그리고 자기가 푼 부분까지는 아무 문제없이 컴파일 되고 실행되게 만들면 부분점수 받음.**

1. **인력관리 프로그램을 작성하라. 주식회사 AOOP에서는 인력을 크게 3분류로 나누며, 각 분류에 속하는 사원의 정보는 아래 표와 같다. (40점)**



* 1. 각 사원의 공통요소를 저장하는 class employee를 구현하고 이를 상속하여 각 분류의 사원 class를 구현하라. 모든 class는 생성자를 가지며, derived class 에서는 base class인 employee의 생성자를 이용하여 정보를 초기화 해야한다. 구현한 class의 유효성을 판단하기 위해 main() 함수에 아래와 같이 3개의 객체를 생성한다. 단, 생성시에는 employee\* list[3]을 기반으로 동적할당을 통해 생성해야한다.(4점)  
     salaried kim(“Kim“, “201-0000“, 3000000, 1000000, “Cho“, “Lee“);  
     permanent lee(“Lee“, “201-0001“, 100000, “Kim“, “Park“);  
     temporary park(“Park“, “201-0002“, 7000, “Kim“, “Lee“);
  2. 각 class의 멤버 변수를 화면에 출력하기 위해서 삽입연산자(<<)를 연산자 오버로딩으로 구현하라. 또한, main()에서 가)에서 선언한 3개의 객채 정보를 << 연산자를 통해 출력하라.(12점)
  3. “emp.dat” 라는 file을 만들어 위에서 선언한 kim, lee, park 객체의 정보를 저장하고, 그 file로부터 각 객체를 읽은 후 각 객채의 정보를 다시 화면에 출력하라. 이 때 “emp.dat” file을 open한 이후 재대로 open이 되었는지 확인하는 분을 try-throw-catch구문을 이용한 exception handling으로 구현하며, open이 잘 이루어지지 않은 경우에는 exit()함수를 호출한다.(12점)
  4. 각 사원의 일년 수입을 계산하는 int annualIncome() 함수를 가상함수로 구현한다. 이 때, employee class는 abstract class가 되어야 한다. 이를 통해 main() 함수에서 세 객체의 연간 수입의 합을 출력하라. 각 사원의 일년 수입의 계산방법은 아래와 같다.(12점)
     1. salaried: 월급\*12+보너스
     2. permanent: 일당\*300
     3. temporary: 시급\*1500

1. **복소수 클래스를 만들어라. (60점)**

복소수는 실수, 허수를 포함하는 수로, a+b*i* 와 같이 표현된다. a와 b는 실수이고 i는 허수단위로 i2 = -1을 만족한다. 실수 a를 그 복소수의 실수부, 실수 b를 복소수의 허수부라고 부른다. 모든 실수는 복소수에 포함된다. 왜냐하면 모든 실수는 허수부가 0인 복소수로 표시할 수 있기 때문이다. 즉 실수 a 는 복소수 a+0i 와 같다. 예를 들어, sqrt 13은 실수부가 sqrt 13이고 허수부가 0인 복소수이다.

복소수 클래스에 추가되어야 할 기능들은 다음과 같다.

* 1. 클래스는 실수부, 허수부 및 아래 기능들을 하는 멤버함수들로 구성 (2점)
  2. Template 클래스로 만들어 실수부, 허수부가 float, double, int 타입이 들어갈 수 있어야 함. (8점)
  3. 아래 코드가 동작하기 위한 constructor들 (7점)  
     complex<double> com1; // 0 + 0i  
     complex<double> com2(3.14); // 3.14 + 0i

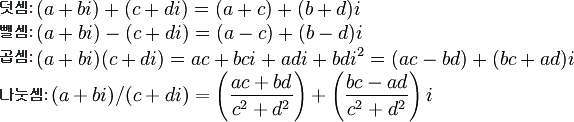
complex<double> com3(1.5, 3.14); // 1.5 + 3.14i

complex<double> com4(com2); // 3.14 + 0i (com2와 동일)

* 1. 아래 코드가 동작하기 위한 assignment operator들 (6점)  
     com1 = com3; // com1: 1.5 + 3.14i

com3 = 2.17; // com3: 2.17 + 0i

* 1. 실수부와 허수부를 리턴해 주는 인터페이스 함수 (2점)
  2. 복소수 클래스간의 +,-,\*,/ 사칙연산을 위한 외부 오퍼레이터 오버로딩 함수 (멤버함수 아님) (12 점)  
     참고로 복소수 사칙연산은 다음과 같이 수행



* 1. 실수와 복소수 클래스간 사칙연산을 위한 오퍼레이터 오버로딩 (참고로 실수는 b가 0인 복소수임)  
     F, G의 결과로 다음과 같은 코드가 실행되어야 함 (8점)  
     cout << com1 + com2 << endl;

cout << com1 - 3.14 << endl;

cout << 2.75 \* com2 << endl;

cout << com4 / com2 << endl;

* 1. 사칙 연산자 중 나눗셈 시 나누기 0 exception을 throw-catch 로 구현하면 추가점수. (+alpha 점)
  2. 단항연산자 오퍼레이터 오버로딩 (com5 = -com1) (2점)
  3. 복소수를 (2.3 + 1.2i) 형식으로 출력해주는 함수.
  4. 복소수의 비교를 위한 ==, != 비교 연산자 오버로딩 (실수부 허수부가 둘 다 같아야 함) (4점)
  5. 복소수의 norm (실수부와 허수부의 제곱의 합) 과 절대값 (norm의 제곱근)을 리턴해 주는 함수 (4점)
  6. 위 기능들이 잘 수행하는지 보여줄 수 있는 main 함수 (이 부분 자체도 점수에 반영) (5점)

2.1 추가로 복소수 3개를 모아놓은 복소수 백터 ((a +bi) , (c +di) , (e +fi))를 구현.

백터클래스의 컨스트럭터, +, - 연산자 오버로딩, 어싸인먼트 오버로딩, 화면 출력함수를 구현  
main에서 적절히 보여줌. (+alpha 점)