1. 택시회사인 “KHU Taxi”에서 택시 운용 수입과 관리 비용을 관리해 주는 SW를 만들어 달라는 의뢰가 들어왔다. 이 회사는 두 종류의 방법으로 택시 및 택시기사를 운영한다. 한 방법은 회사는 기사에게 월급을 주고 기사는 수익을 모두 회사에 납입하는 방식이다. 두 번째는 매일 기사의 수입에서 30%의 수수료를 떼는 방식이다. 두 종류 모두 택시는 회사에서 관리하여 관리비용이 나간다. 다음의 클래스들을 만들고 이들 사이의 관계를 적절히 구성하라. (30점)
   1. DriverManager 클래스
      1. 택시기사 고객 관리 클래스. 여기에는 다수의 “EmployedDriver” 와 “CommisionDriver”에 대한 데이터를 저장할 수 있어야 함
      2. 택시기사를 추가할 수 있는 기능이 있어야 함
      3. 저장된 기사들에게 받는 수익을 계산할 수 있어야 함
      4. 저장된 기사들에게 지출되는 비용을 계산할 수 있어야 함 (인건비).
      5. 기사들에게 딸린 택시들의 유지비용을 계산할 수 있어야 함.
   2. EmployedDriver 클래스
      1. 월급 주고 수익을 납부하는 형태의 택시기사.
      2. 이름, 월급, 일간 평균 승객 수, 승객당 평균 택시비를 저장할 수 있어야 함.
      3. 한달 수익 (평균승객수\*승객당 평균택시비\*30) 계산할 수 있어야 함.
      4. 월별 비용 (기사 월급 + 택시 운영비용)을 계산할 수 있어야 함.
      5. 이 운전사가 운전하는 택시 한대를 저장할 수 있어야 함.
   3. CommisionDriver 클래스
      1. 30%의 수수료를 납부하는 택시기사
      2. 이름, 일간 평균 승객 수, 승객당 평균 택시비를 저장할 수 있어야 함.
      3. 한달 수익 (승객수\*평균택시비\*0.3\*30)를 계산 할 수 있어야 함.
      4. 월별 비용 (택시 운영비용)을 계산할 수 있어야 함.
      5. 이 운전사가 운전하는 택시 한대를 저장할 수 있어야 함.
   4. Taxi클래스
      1. 이름, 월별 운영 비용을 저장할 수 있어야 함.
   5. 다음 Main 이 동작하게 위 클래스들의 내용을 적절히 만들어라. 필요하다면 맴버함수, 변수추가 가능
   6. EmployedDriver와 CommisionDriver는 공통적인 맴버변수와 맴버함수 몇 가지를 가진다. “상속” 이라는 기능으로 코드를 좀 더 효율적으로 만들 수 있다. (힌트: 공통된 부분을 묶어주는 공통 부모 클래스인 Driver 클래스). 이를 수행하라. 그리고, 아래 main 코드를 분석해 보면 이 상속관계는 단순 상속관계가 아니고, Driver는 “순수가상함수”를 가지는 “추상클래스”로 만들어야 할 것 같다. 그렇게 만들어라. (힌트: 공통으로 가지는 함수는 순수가상함수로 만들기)
   7. **또 다른 힌트 힌트: DriverManager 에서 EmployedDriver와 CommisionDriverBusdml 리스트를 따로 관리하지 않고 다형성을 이용해 Driver의 리스트로 관리하면 아래 Main 가능!! 다형성을 적용하려면 리스트는 Driver \*의 배열이 되어야 함 (최대 Driver 수는 100개라고 가정)**
   8. 다음 Main 이 동작하게 만들어라.

DriverManager KHUDrivers;

Driver \* temp = new employedDriver(“홍길동”,2500000, 54,7500, Taxi(“천리마,300000)); //이름, 윌급, 일별승객수, 승객별 택시비, 택시(이름,유지비용)

KHUDrivers.addDriver(temp); //temp 가 가리키는 기사 추가

… //최소 5명 이상의 기사 추가

temp = new CommisionDriver(“캔디”,43, 8000, Taxi(“붕붕”, 250000)); //이름, 일별승객수, 승객별 택시비, 택시

KHUDrivers.addDriver(temp);

cout << “한달수입: “ << KHUDrivers.calculateTotalIncomes() << endl; //한달 총 수입: 모든 기사에 대해 회사의 한달 간 수익 합

cout << “한달지출: “ << KHUDrivers.calculateTotalExpenses() << endl; //한달 총 지출: 모든 기사에 대해 회사의 한달 간 지출 합

1. 위 택시회사가 승승장구하여 SW개발쪽으로 사업을 확장하려고 한다. 이 회사에서 첫 번째 출시할 제품은 카카오톡을 물리칠 가칭 “KHUKHUO-TALK” 이다. 당신은 지금 이 회사의 새로운 SW개발 부서의 개발 책임자이고, 이 쿠쿠오톡 앱을 객체지향 개념을 기반으로 디자인하려고 한다. 우선은 이 쿠쿠오톡의 기능적인 측면은 카카오톡과 똑같다고 가정하자. 이 상황에서 여러분이 필요하다고 생각되는 클래스들과 이들의 관계를 디자인 하라. (10점)
   1. 코딩으로 구현할 필요는 없고 글로 cpp 파일에 comment 로 작성
   2. 사용자 스마트폰에 인스톨 될 앱에 대한 디자인임 (쿠쿠오 톡 서버 아님)
   3. 이 앱에서 필요한 객체가 어떤게 있을지 (사용자, 채팅방 등등)에 대해 생각해 보고 이를 위주로 디자인
   4. 앱의 세부 기능 및 세부 스펙에 대한 정의가 필요하다고 생각되면 정의해도 됨
   5. 각 클래스의 역할, 대략적인 데이터 및 함수, 각 클래스간의 관계 (상속, Aggregation 등) 에 대한 기술 필요
   6. 정답은 없지만 완성도 및 적절성으로 평가할 예정
2. 복소수 클래스를 만들어라. (40점)

복소수는 실수, 허수를 포함하는 수로, a+bi 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. (소스코드에 아래 몇 번까지 구현했는지 표시할 것)

* 1. 클래스는 실수부, 허수부 및 아래 기능들을 하는 멤버함수들로 구성 (1점)
  2. Template 클래스로 만들어 실수부, 허수부가 float, double, int 타입이 들어갈 수 있어야 함. (6점)
  3. 아래 코드가 동작하기 위한 constructor들 (5점)

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)

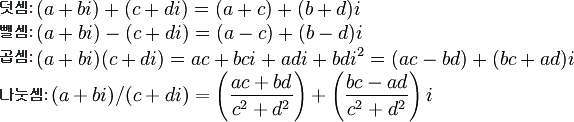
* 1. 아래 코드가 동작하기 위한 assignment operator들 (4점)

com1 = com3; // com1: 1.5 + 3.14i

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

* 1. 실수부와 허수부를 리턴해 주는 인터페이스 함수 (1점)
  2. 복소수 클래스간의 +,-,\*,/ 사칙연산을 위한 외부 오퍼레이터 오버로딩 함수 (멤버함수 아님) (10 점)

참고로 복소수 사칙연산은 다음과 같이 수행



* 1. 실수와 복소수 클래스간 사칙연산을 위한 오퍼레이터 오버로딩 (참고로 실수는 b가 0인 복소수임)
  2. F, G의 결과로 다음과 같은 코드가 실행되어야 함 (6점)

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. (4 점)
  2. 복소수를 (2.3 + 1.2i) 형식으로 출력해주는 함수.
  3. 위 기능들이 잘 수행하는지 보여줄 수 있는 main 함수 (이 부분 자체도 점수에 반영) (3점)

1. 복소수 클래스를 3개 묶어 놓은 복소수 벡터를 만들고 이를 바이너리 파일 입출력하라. (20점)
   1. 복소수 3개를 모아놓은 복소수 벡터 ComplexVector를 구현하라. 복소수 벡터는 ((a +bi) , (c +di) , (e +fi)) 과 같이 표현 가능함. (3번을 다 풀지 못했어도 간단한 complex 클래스를 만들어서 수행해도 됨)
   2. 위의 ComplexVector 클래스를 이용해 객체 10개를 만들어라.
   3. 다음 순서로 파일 입출력하라.
      1. 바이너리 출력 file open
      2. ComplexVector 형 v1, v2, v3 만들어 임의의 수로 채우고 화면에 출력
      3. binary IO로 v1, v2, v3을 위 파일로 출력한 다음 파일 close
      4. 위의 파일 바이너리 읽기 모드로 다시 오픈
      5. ComplexVector형 answer 객체 하나 만듦
      6. 1에서 3까지의 숫자 하나를 cin으로 사용자 입력 받음 (입력 범위 벋어나는 Exception을 C++예외처리로 처리)
      7. seek 함수를 이용해 입력 받은 숫자의 순서에 해당하는 벡터를 파일로부터 읽어 answer에 저장. 예) 3을 입력받으면 3번째 백터 (v3)읽음.
      8. answer 출력