

DEPT: _____ SID: _____ Name: _____

실습시험 문제(100 점, 60 분):

1. 아래 선언된 **Complex** 클래스 형의 두 객체 **a, b**에 대해 **a + b**의 연산이 가능하도록 연산자 오버로딩(operator overloading)을 수행하려고 한다.(50pts) (프로그램 구현 능력 평가)

```

1 // Lab 1: Complex.h
2 #ifndef COMPLEX_H
3 #define COMPLEX_H
4
5 class Complex
6 {
7 public:
8     Complex( double = 0.0, double = 0.0 ); // default constructor
9     Complex add( const Complex & ); // function add
10    Complex subtract( const Complex & ); // function subtract
11    void printComplex(); // print complex number format
12    void setComplexNumber( double, double ); // set complex number
13 private:
14    double realPart;
15    double imaginaryPart;
16 }; // end class Complex
17
18 #endif

```

- (1) 임의의 자료형(arbitrary types)을 사용할 수 있도록 클래스를 다시 선언하시오. (10pts)
- (2) 전역 함수 (global function)의 형식으로 '+' 연산자 오버로딩을 구현하시오.(15pts)
- (3) 클래스 멤버 함수 (class member function)의 형식으로 '-' 연산자 오버로딩을 구현하시오.(15pts)
- (4) $(15.0+20.3i) + (5.5+7.6i)$ 와 $(50+10i) - (40+55i)$ 연산을 하고 해당결과를 출력할 수 있는 main함수를 작성하시오.(10pts)

Ans:

DEPT: _____ SID: _____ Name: _____

2. 주어진 소스코드를 이용하여 아래의 물음에 답하시오.(코드파악 능력 및 코딩능력 평가).

- (1) 아래의 프로그램에서 “____”로 표시된 빈자리를 채우시오 {(1)~(10)}. (20pts = 2pts x 10)
- (2) 각 클래스에 대한 UML 다이어그램을 그리시오.(10pts)
- (3) 주어진 결과와 동일한 결과를 출력할수 있는 main함수를 main.cpp에 작성하시오. (15pts)
(main 함수에서 기본(base)클래스의 핸들(포인터)를 이용하여 파생클래스 객체를 처리할것).
- (4) 클래스간의 상속관계를 나타내는 관계도(inheritance hierarchy)를 그리시오. (5pts)

```
//Book.h
#ifndef BOOK_H
#define BOOK_H
#include <iostream>
using namespace std;
#include <string>
using std::string;
class Book{
private:
    string book;
public:
    Book(string);
    void setbook(string);
    string getbook();
    (1) void print() const;
    virtual ~Book();
};
#endif
```

```
//Book.cpp
#include "Book.h"
#include <iostream>
using namespace std;
#include <iomanip>
Book::Book(string b){
    setbook(b);
    cout<<"Base Book - Constructor for "<<getbook()<<endl;
}
void Book::setbook(string b){
    book = b;
}
(2) Book::getbook(){
    return book;
}
void Book::print() const{
    cout<<"Book"<<endl;
}
Book::~~Book(){
    cout<<"Base Book - Destructor for "<<getbook()<<endl;
}
```

```
//Novel.h
#ifndef NOVEL_H
#define NOVEL_H
(3)
#include <iostream>
using namespace std;
```

2021 Final Exam of Applications of OOP (ICE2013)

DEPT: _____ SID: _____ Name: _____

```
class Novel : public Book{
public:
    Novel(string);
    void print() const;
    ~Novel();
};
#endif
```

```
//Novel.cpp
_____ (4)
#include <iostream>
using namespace std;
#include <iomanip>
Novel::Novel(string novel)
    :Book(novel)
{
    _____ (5)
    cout<<"derived Novel- Constructor for "<<getbook()<<endl;
}
void Novel::print() const{
    cout<<"Novel"<<endl;
}
Novel::~Novel(){
    cout<<"derived Novel- Destructor "<<getbook()<<endl;
}
```

```
//Biography.h
#ifndef BIOGRAPHY_H
#define BIOGRAPHY_H
_____ (6)
#include <iostream>
using namespace std;
class Biography : public Book{
public:
    Biography(string);
    void print() const;
    ~Biography();
};
#endif
```

```
//Biography.cpp
_____ (7)
#include <iostream>
using namespace std;
#include <iomanip>
Biography::Biography(string Bio)
    :Book(Bio)
{
    _____ (8)
}
void Biography::print() _____ (9){
    cout<<"Biography"<<endl;
}
Biography::~Biography()
{
    _____ (10)
}
```

2021 Final Exam of Applications of OOP (ICE2013)

DEPT: _____ SID: _____ Name: _____

//main.cpp

실행결과

```
C:\Windows\system32\cmd.exe
Base Book - Constructor for The Alchemist
derived Novel- Constructor for The Alchemist
Base Book - Constructor for Paulo Coelho
derived Biography- Constructor for Paulo Coelho
Novel
Biography
derived Novel- Destructor The Alchemist
Base Book - Destructor for The Alchemist
derived Biography- Destructor Paulo Coelho
Base Book - Destructor for Paulo Coelho
계속하려면 아무 키나 누르십시오 . . .
```

Ans: