

Soal 1. Inheritance, Abstract Class/Function

a. Lengkapi header kelas abstrak `Hotel` berikut ini dengan prototype (tanpa membuat *inline implementation*) sesuai dengan spesifikasi yang dituliskan pada komentar. Gunakan nama fungsi/prosedur sesuai yang diberikan di spesifikasi. Tambahkan keyword `virtual`, `const`, dan/atau `static` sesuai dengan kaidah yang diajarkan untuk menjamin kode Anda benar. Jika anggota kelas tidak perlu direalisasikan, tuliskanlah:

// TIDAK PERLU (Anda juga tidak perlu menuliskan *prototype*-nya).

```
// Class Hotel
// File: Hotel.h

#ifndef HOTEL_H
#define HOTEL_H

#include <string>
using namespace std;

class Hotel {
protected:
    // Atribut
    string name;
    string bintang; //Melati; bintang_tiga; bintang_empat; bintang_lima
    int openYear;
public:
    // User-defined constructor: set nilai atribut berdasarkan nilai parameter masukan

    // Default constructor: set nilai atribut sbb:
    // name = "noname"; openYear = 1900; bintang = "none"

    // Copy constructor

    // Destructor

    // Operator Assignment

    // ... set_name(...)

    // ... get_name()

    // ... set_bintang(...)

    // ... get_bintang()

    // ... get_age(), asumsikan tahun saat ini dapat diakses dengan makro CURRENT_YEAR

    // ... displayInfo(): Mencetak nama, umur hotel, bintang, dan room_rate

    // ... rate(): menghitung biaya menginap sesuai dengan umur dan tergantung bintang hotel

};

#endif // HOTEL_H
```

- b. Buatlah *subclass* dengan mewariskan kelas `Hotel` pada kelas `bintang_empat` dan `bintang_lima`, dengan ketentuan sebagai berikut (buat *overload function* jika diperlukan):
- Kelas `bintang_empat` mencatat jumlah star layanan setiap hotel bintang empat, diakses menggunakan `set_star()` dan `get_star()`. *Override* `displayInfo()` untuk mencetak informasi tambahan yaitu `totalRate` yang didapatkan dari perkalian antara `rate` dengan star.
 - Kelas `bintang_lima` mencatat jumlah facility yang sudah digunakan (`set_facility()`, `get_facility()`) dan memiliki fungsi `calculateFacility()` yang menghasilkan `expense` yaitu jumlah facility dikalikan dengan 100000. *Override* `displayInfo()` untuk mencetak informasi tambahan yaitu `expense`.
- c. Tuliskan implementasi kelas abstrak `Hotel` dalam `Hotel.cpp` sesuai dengan spesifikasi pada header file.
- d. Lengkapi `main.cpp` berikut pada bagian yang kosong untuk menguji perilaku kelas-kelas yang Anda buat dengan:

```
#include "Hotel.h"
#include <iostream>
using namespace std;

int main() {
    _____ aston = bintang_empat("Hotel Aston", 2010);
    _____ padma = bintang_lima("Hotel Padma", 2000);

    // cetak semua hotel dan informasi semua hotel
    // [gunakan Hotel::displayInfo()]

    _____
    _____

}
```

Soal 2. Generic Class dan Exception Handling

Berikut ini adalah contoh serangkaian aksi enqueue dan dequeue pada sebuah queue dengan elemen bertipe integer dan memiliki kapasitas 3 elemen.

No.	Aksi	Kondisi Queue
0	Kondisi awal kosong	Elemen:
1	dequeue()	Elemen: Queue telah kosong
2	enqueue(8)	Elemen: 8
3	enqueue(6)	Elemen: 8, 6
4	enqueue(4)	Elemen: 8, 6, 4
5	enqueue(9)	Elemen: 8, 6, 4 Queue telah penuh
6	dequeue(), returns 8	Elemen: 6, 4
7	enqueue(2)	Elemen: 6, 4, 2
8	dequeue(), returns 6	Elemen: 4, 2

Sebuah struktur data Queue dapat dibuat generik sehingga elemennya dapat bertipe apapun.

Tugas kalian adalah:

- Dengan bahasa C++, buatlah sebuah kelas Queue generik dengan kapasitas maksimal 3 elemen. Kelas tersebut minimal harus memiliki:

- a. Default constructor.
 - b. Destructor.
 - c. Method enqueue, untuk memasukkan sebuah elemen ke dalam Queue. Method ini akan melempar sebuah exception jika Queue sudah penuh.
 - d. Method dequeue, untuk mengeluarkan sebuah elemen ke dalam Queue. Method ini akan melempar sebuah exception jika Queue kosong.
 - e. Operator overloading << untuk menampilkan isi Queue (bentuk tampilan bebas).
2. Membuat kelas abstrak QueueException dengan method what() yang mengembalikan informasi tipe exception. Kelas turunan QueueException dapat menangani 2 jenis exceptions:
 - a. Suatu Queue telah kosong.
 - b. Suatu Queue telah penuh.
3. Tulislah main program yang membuat sebuah Queue dengan isi seperti contoh di atas. Gunakanlah exception handler untuk menangani exception yang dilemparkan oleh kelas Queue.
4. Tunjukkan bahwa struktur data Queue telah dibuat generik dengan mampu menampung elemen yang beragam:
 - a. Minimal 2 tipe data "built-in", yakni Queue of int dan Queue of float.
 - b. Minimal 1 tipe data "customized", yakni Queue of Mobil (definisi kelas Mobil dibebaskan).

Gunakan kerangka program berikut ini. Program dikerjakan dalam satu buah file main.cpp untuk sekedar alasan kepraktisan dalam pemeriksaan ujian. Jika ada bagian kerangka program yang kurang lengkap atau salah, Anda dapat melengkapinya atau menggantinya.

```
#include <iostream>

using namespace std;

template <class T>
class Queue {
private:
    // lengkapi private member di sini
    T *elements;
    ...
public:
    // lengkapi public member di sini
    ...
};

// Default constructor
// Destructor
// Method enqueue
// Method dequeue
// Operator overloading <<
```

```
// Main function
int main() {

}
```

Soal 3. STL

Di negara NeverHeardBefore, pemimpin pemerintahan memiliki masa jabatan yang ditentukan oleh voting. Seorang pemimpin memiliki nilai diri tertinggi dibandingkan dengan birokrat lainnya. Setiap pemimpin/ birokrat memiliki dua birokrat lainnya untuk membantu, dengan ketentuan nilai diri mereka lebih tinggi dari dua asistennya tersebut. Struktur seperti ini dikenal dengan max heap yang merupakan implementasi dari priority queue. Jika pemimpin sekarang di-vote untuk berhenti memimpin, maka penggantinya diambil dari salah satu asisten yang memiliki skor tertinggi. Gunakan STL untuk mengimplementasikan hal berikut:

1. Membuat sebuah heap yang terdiri dari 50 bilangan integer random untuk merepresentasikan struktur pemimpin/ birokrat negara NeverHeardBefore
2. Menambahkan birokrat baru dengan nilai diri integer random
3. Melakukan vote terhadap pemimpin (menghapus elemen maksimum dari heap)
4. Mengurutkan struktur birokrat/ heap tersebut.

Note: kalau butuh contoh heap http://www.cplusplus.com/reference/queue/priority_queue/

atau dapat juga memanfaatkan vector