

ITB_IF2010_2_2425 Pemrograman Berorientasi Objek

[Dashboard](#) / [My courses](#) / [ITB_IF2010_2_2425](#) / [Praktikum 2](#) / [Praktikum 2 \(Latihan\)](#)

Started on	Wednesday, 26 March 2025, 9:30 PM
State	Finished
Completed on	Thursday, 27 March 2025, 1:16 AM
Time taken	3 hours 45 mins
Grade	300.00 out of 300.00 (100%)

Question **1**
Correct
Mark 100.00 out of 100.00
Flag question

Time limit	1 s
Memory limit	64 MB

Pada praktikum ini, Anda diminta untuk mengimplementasikan **kelas generik KeyValue** dalam bahasa pemrograman C++. Kelas ini akan menyimpan pasangan **key** dan **value** dengan beberapa batasan dan aturan khusus.

Spesifikasi Kelas KeyValue

Atribut:

1. **keys**: array yang menyimpan **key** dengan panjang maksimum **10**.
2. **values**: array yang menyimpan **value** yang berkorespondensi dengan **key** pada indeks yang sama, dengan panjang maksimum **10**.
3. Tipe data **key** dan **value** bisa bertipe apapun, baik sama maupun berbeda.

Metode:

1. **set(key, value)**: Menerima dua parameter bertipe sesuai dengan **key** dan **value**. Jika **key** sudah ada dalam array, maka **value** diperbarui. Jika array penuh, cetak **"KeyValue penuh! Tidak bisa menambahkan KeyValue lagi."** diakhiri dengan **newline**.
2. **display(key)**: Menampilkan **value** yang sesuai dengan **key**. Jika **key** tidak ditemukan, cetak **"Key tidak ditemukan!"** diakhiri dengan **newline**.

Kasus Khusus:

1. Jika **key** bertipe **double**, pencocokan dilakukan dengan toleransi **kurang dari 0.01**.
2. Contoh: **1.01** dan **1.012** dianggap **sama**, tetapi **1.01** dan **1.02** dianggap **berbeda**.

Contoh Penggunaan

```
```cpp
KeyValue<int, string> kv;
kv.set(1, "Alice");
kv.set(2, "Bob");
kv.set(3, "Charlie");
kv.display(1); // Output: "Alice"
kv.display(2); // Output: "Bob"
kv.display(3); // Output: "Charlie"
```
```

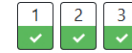
NOTE

1. Jangan lupa untuk mendefinisikan constructor, copy constructor, destructor, dll yang sekiranya penting untuk digunakan
2. Kumpulkan satu file bernama **KeyValue.cpp**

C++14

KeyValue.cpp

Quiz navigation

[Show one page at a time](#)[Finish review](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 10 | Accepted | 0.00 sec, 2.96 MB |
| 2 | 10 | Accepted | 0.00 sec, 2.99 MB |
| 3 | 10 | Accepted | 0.00 sec, 2.96 MB |
| 4 | 20 | Accepted | 0.00 sec, 3.07 MB |
| 5 | 10 | Accepted | 0.00 sec, 2.89 MB |
| 6 | 10 | Accepted | 0.00 sec, 2.89 MB |
| 7 | 10 | Accepted | 0.00 sec, 2.95 MB |
| 8 | 10 | Accepted | 0.00 sec, 3.04 MB |
| 9 | 10 | Accepted | 0.00 sec, 3.03 MB |

Question **2**

Correct

Mark 100.00 out of 100.00

Flag question

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Di tengah malam yang gelap, markas rahasia **O.W.C.A.** tiba-tiba bergetar akibat alarm darurat. Mayor Monogram muncul di layar besar dengan ekspresi serius.

"Agen Rahasia! Kami dalam keadaan darurat! Dr. Asep Spakbor kembali dengan rencana jahatnya. Kali ini, dia telah menciptakan makhluk buas yang dapat menghancurkan pertahanan kita!"

Tepat saat itu, **Purry**, agen terbaik di O.W.C.A., melompat ke meja dengan penuh semangat.

*"Benar, Purry. Kami membutuhkan bantuanmu! Dr. Asep Spakbor telah membangkitkan Naga Legendaris yang hanya bisa dikalahkan jika kita memahami cara kerja makhluk ini. Kami telah memperoleh blueprint dari dua makhluk rahasia: **Creature** dan **Dragon**."*

Mayor Monogram lalu menyerahkan dua dokumen rahasia kepada Anda:

- **Creature.hpp** → Menjelaskan dasar dari makhluk-makhluk misterius.
- **Dragon.hpp** → Memberikan rincian tentang naga yang diciptakan oleh Dr. Asep Spakbor.

"Tugasmu," lanjut Mayor Monogram, "adalah memanfaatkan kedua blueprint ini dan memahami bagaimana mereka bekerja. Hanya dengan itu, kita dapat menghadapi Naga Legendaris dengan strategi yang benar!"

Lengkapi **main.cpp** agar dapat menghasilkan **output yang diharapkan**.

Kamu **harus** memanfaatkan kedua kelas yang telah diberikan (**Creature.hpp** dan **Dragon.hpp**) tanpa mengubah isi dari file tersebut.

PERHATIAN:

- **Tidak boleh** menambah atau mengurangi **#include** pada **main.cpp**.
- Gunakan **polymorphism** dan **inheritance** dengan benar.

File main.cpp Lengkapi bagian kosong berikut agar O.W.C.A. dapat memahami makhluk ini sepenuhnya:

```
#include "Creature.hpp"
#include "Dragon.hpp"
```

```
int main() {
    // Lengkapi disini

    return 0;
}
```



Output yang diharapkan

```
A creature has been born!
A dragon has been born!
I am the mighty dragon Draco! Beware of my flames!
Base Health: 100; Additional Health: 80
I am the mighty dragon Draco! Beware of my flames!
Base Health: 90
Base Health: 90; Additional Health: 80
I am a mystical creature named Draco!
Base Health: 80
Base Health: 90; Additional Health: 80
```



Dapatkah kamu menyelesaikan ujian rahasia dari O.W.C.A. ini dan membuktikan bahwa kamu adalah agen terbaik? 🗨️ 🔍 🚀

C++14

 main.cpp

Score: 100

Blackbox

Score: 50

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 50 | Accepted | 0.00 sec, 2.97 MB |

Whitebox

Score: 50

| No | Score | Checker | Description |
|----|-------|----------------|-------------|
| 1 | 50 | Custom checker | CORRECT |

Question **3**

Correct

Mark 100.00 out of 100.00

 Flag question


| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Ticket Manager


Sistem Ticket Manager digunakan untuk mengelola penjualan tiket suatu acara.

Terdapat sebuah base class **Ticket**, dengan **RegularTicket** dan **VIPTicket** sebagai turunannya.

Implementasi untuk kelas tiket sudah tersedia.


 File yang Disediakan (TicketManagerToolkit.zip)

- **Ticket.hpp**, **RegularTicket.hpp**, **VIPTicket.hpp**
- **Ticket.cpp**, **RegularTicket.cpp**, **VIPTicket.cpp**
- **TicketManager.hpp** (Berisi penjelasan lengkap tentang fungsi yang perlu diimplementasikan)
- **test.cpp** (Untuk menguji implementasi)

 Tugas Anda

- Implementasikan **TicketManager.cpp** sesuai spesifikasi pada **TicketManager.hpp**.
- Seluruh output (hasil print) harus diakhiri dengan **endl**.
- Kumpulkan **TicketManager.cpp**

C++14

 TicketManager.cpp

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 10 | Accepted | 0.00 sec, 3.01 MB |
| 2 | 10 | Accepted | 0.00 sec, 2.96 MB |
| 3 | 10 | Accepted | 0.00 sec, 3.16 MB |
| 4 | 10 | Accepted | 0.01 sec, 3.06 MB |

| | | | |
|----|----|----------|-------------------|
| | | | |
| 5 | 10 | Accepted | 0.00 sec, 3.00 MB |
| 6 | 10 | Accepted | 0.00 sec, 2.96 MB |
| 7 | 10 | Accepted | 0.00 sec, 3.05 MB |
| 8 | 10 | Accepted | 0.00 sec, 3.07 MB |
| 9 | 10 | Accepted | 0.00 sec, 2.97 MB |
| 10 | 10 | Accepted | 0.00 sec, 3.01 MB |

[Finish review](#)

[◀ Praktikum 2](#)

Jump to...



You are logged in as [13523014 Nicholas Andhika Lucas](#) ([Log out](#))

[ITB_IF2010_2_2425](#)

[Data retention summary](#)

[Get the mobile app](#)