物件導向程式設計

Template (2/3)

Joseph Chuang-Chieh Lin
Dept. CSIE, Tamkang University

Platform

- Dev-C++

  Click here to download.

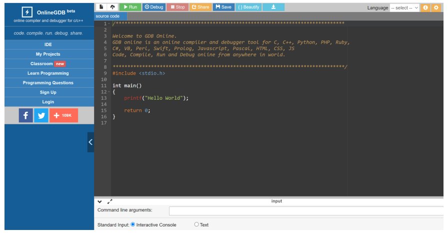
  Note: Please use this version otherwise you can't compile your programs/projects in Win10.

Open Source C/C++ IDE for Windows

Dev-C++ is a full-featured C and C++ Integrated Development Environment (IDE) for Windows platforms. Millions of developers, students and researchers use Dev-C++ since the first version was released in 1998. It has been featured in dozens of C++ and scientific books and remains one of the favorite learning tool among universities & schools worldwide.

Get Dev-C++

- OnlineGDB (https://www.onlinegdb.com/)

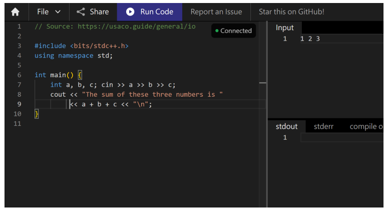


```
/******************************************************************************

Welcome to GDB Online.
GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby,
C#, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML, CSS, JS
Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************************/
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Command line arguments:

Standard Input: [✓ Interactive Console] Text

- Real-Time Collaborative Online IDE (https://ide.usaco.guide/)



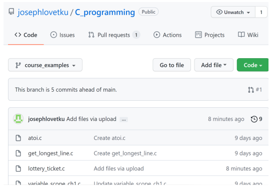
```
// Source: https://usaco.guide/general/io

#include <bits/stdc++.h>
using namespace std;

int main() {
    int a, b, c; cin >> a >> b >> c;
    cout << "The sum of these three numbers is "
         << a + b + c << "\n";
}
```

Input:

```
1 2 3
```

- Other resources:
  - MIT OpenCourseWare - Introduction to C++ [link].
  - Learning C++ Programming [Programiz].
  - GeeksforGeeks [link]

My GitHub page:
click the link here to visit.



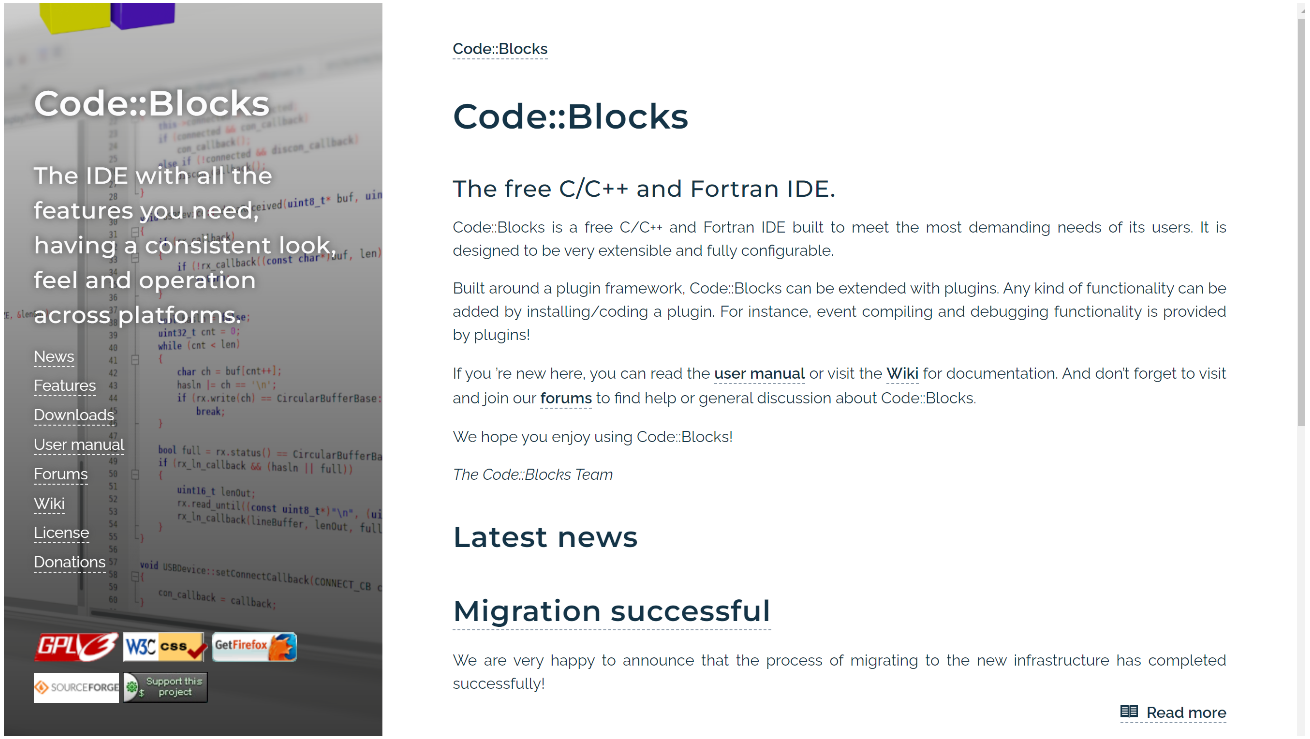
This branch is 5 commits ahead of main. #1

josephlovetku Add files via upload ... 8 minutes ago 9

atoi.c	Create atoi.c	9 days ago
get_longest_line.c	Create get_longest_line.c	9 days ago
lottery_ticket.c	Add files via upload	8 minutes ago
variable_scope_ch1.c	Update variable_scope_ch1.c	9 days ago

Platform/IDE

- https://www.codeblocks.org/



Code::Blocks
The IDE with all the features you need, having a consistent look, feel and operation across platforms.
News
Features
Downloads
User manual
Forums
Wiki
License
Donations
Code::Blocks

Code::Blocks

The free C/C++ and Fortran IDE.

Code::Blocks is a free C/C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Built around a plugin framework, Code::Blocks can be extended with plugins. Any kind of functionality can be added by installing/coding a plugin. For instance, event compiling and debugging functionality is provided by plugins!

If you 're new here, you can read the user manual or visit the Wiki for documentation. And don't forget to visit and join our forums to find help or general discussion about Code::Blocks.

We hope you enjoy using Code::Blocks!

The Code::Blocks Team

Latest news

Migration successful

We are very happy to announce that the process of migrating to the new infrastructure has completed successfully!

Combining operator overloading

```
struct Node {
    int data;
    int order; // the order of generation
    Node *next;
    Node() { // constructor }
    ~Node() { // destructor }
    static int counter;
    // overloading '<' and '<<'
};

int Node::counter = 0; // total #objects
```

```
template <class T>
void bubbleSort(T a[], int n) {
    for (int i=0; i<n-1; i++)
        for (int j=n-1; i<j; j--)
            if (a[j] < a[j-1])
                swap(a[j], a[j-1]);
}
```

```
int main() {
    int n = 5;
    Node *dataList = new Node[n];
    bubbleSort<Node>(dataList, n);
    for (int i = 0; i < n; i++)
        cout << dataList[i] << " ";
    cout << endl;
    delete [] dataList;
    return 0;
}
```

Sample input :

```
10 20 -5 77 29
```

Sample output :

```
2:-5 0:10 1:20 4:29 3:77
```

More Arguments to Templates

```
#include <iostream>
using namespace std;

template <class T, class U> class A {
    T x;
    U y;
public:
    A() { cout << "Constructor Called\n"; }
    A(T a, U b): x(a), y(b) {
        cout << x << ", " << y << endl;
    }
};

int main() {
    A<char, char> a;
    A<int, double> b;
    A<int, char>c(100, 'T');
    return 0;
}
```

Example: A Generic Array

```
template <class T, int size>
class Array {
private:
    T myArr[size];

public:
    Array(T arr[]);
    void print();
};
```

```
template <class T, int size>
void Array<T,size>::print() {
    int i = 0;
    for (int i = 0; i < s; i++)
        cout << " " << myArr[i];
    cout << endl;
}
```

```
template <class T, int size>
Array<T,size>::Array(T arr[]) {
    ptr = new T[size];
    for (int i = 0; i < size; i++)
        myArr[i] = arr[i];
}
```

```
int main() {
    int arr[3] = { 1, 2, 3};
    Array<int, 3> a(arr, 3);
    a.print();
    return 0;
}
```

Remark

- Both function overloading and templates are examples of polymorphism of OOP.
  - *Function overloading*: multiple functions do similar tasks.
  - *Templates*: multiple functions do **identical** tasks.

Inheritance of a Template

```
template<typename T>
class Base {
public:
    Base(T data): mData(data) { }
    virtual void print() {
        cout << mData << endl;
    }
protected:
    T mData;
};
```

```
template<typename T>
class Derived : public Base<T> {
public:
    Derived() = default;
    Derived(T data) : Base<T>(data) { }

    void print() override {
        cout << "data = "
             << this->mData << endl;
    }
};
```

```
int main() {
    Derived<float> d1(5.2f);
    Derived<std::string> d2("TKU_CSIE");
    d1.print();
    d2.print();
    return 0;
}
```

Exercise

- Please modify the following class `List` so that the main function can run successfully.

```
#include <iostream>
using namespace std;

class List {
public:
    List() : head_(nullptr) { }
    virtual void add(int n) {
        Link *p = new Link(n, head_);
        head_ = p;
    }
    void print_head() {
        cout << "head: " << head_->val << endl;
    }
private:
    struct Link {
        int val;
        Link *next;
        Link(int n, Link* nxt): val(n), next(nxt) { }
    };
    Link * head_;
};
```

```
int main() {
    List<int> nums;
    nums.add(1);
    nums.add(2);
    nums.print_head();
    return 0;
}
```

Sample output

```
head: 2
```