Hello world

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

    cout<<"Hello, World!";

    return 0;

}

Input and output

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

    /\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

    int a,b,c;

    cin>>a>>b>>c;

    cout<<a+b+c;

    return 0;

}

Basic datatypes

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

    // Complete the code.

    int a;

    long b;

    char c;

    float d;

    double e;

    scanf("%d %ld %c %f %lf",&a,&b,&c,&d,&e);

    printf("%d\n%ld\n%c\n%f\n%lf",a,b,c,d,e);

    return 0;

}

Conditional statements

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

int main()

{

    string n\_temp;

    getline(cin, n\_temp);

    int n = stoi(ltrim(rtrim(n\_temp)));

    // Write your code here

    cin>>n;

    if(n==1)

    {

        cout<<"one";

    }

    else if(n==2)

    {

        cout<<"two";

    }

    else if(n==3)

    {

        cout<<"three";

    }

    else if(n==4)

    {

        cout<<"four";

    }

    else if(n==5)

    {

        cout<<"five";

    }

    else if(n==6)

    {

        cout<<"six";

    }

    else if(n==7)

    {

        cout<<"seven";

    }

    else if(n==8)

    {

        cout<<"eight";

    }

    else if(n==9)

    {

        cout<<"nine";

    }

    else

    {

        cout<<"Greater than 9";

    }

    return 0;

}

string ltrim(const string &str) {

    string s(str);

    s.erase(

        s.begin(),

        find\_if(s.begin(), s.end(), not1(ptr\_fun<int, int>(isspace)))

    );

    return s;

}

string rtrim(const string &str) {

    string s(str);

    s.erase(

        find\_if(s.rbegin(), s.rend(), not1(ptr\_fun<int, int>(isspace))).base(),

        s.end()

    );

    return s;

}

For loop

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

    // Complete the code.

    int a,b;

    cin>>a>>b;

    for(int i=a; i<=b; i++)

    {

        if(i<10)

        {

            if(i==1)

            {

                cout<<"one"<<endl;

            }

            else if(i==2)

            {

                cout<<"two"<<endl;

            }

            else if(i==3)

            {

                cout<<"three"<<endl;

            }

            else if(i==4)

            {

                cout<<"four"<<endl;

            }

            else if(i==5)

            {

                cout<<"five"<<endl;

            }

            else if(i==6)

            {

                cout<<"six"<<endl;

            }

            else if(i==7)

            {

                cout<<"seven"<<endl;

            }

            else if(i==8)

            {

                cout<<"eight"<<endl;

            }

            else if(i==9)

            {

                cout<<"nine"<<endl;

            }

        }

            else

            {

                if(i%2==0)

                {

                    cout<<"even"<<endl;

                }

                else

                {

                    cout<<"odd"<<endl;

                }

            }

    }

    return 0;

}

Functions

#include <iostream>

#include <cstdio>

using namespace std;

/\*

Add `int max\_of\_four(int a, int b, int c, int d)` here.

\*/

int max\_of\_four(int a, int b, int c, int d)

{

    if(a>b && a>c && a>d)

    {

        return a;

    }

    else if(b>a && b>c && b>d)

    {

        return b;

    }

    else if(c>a && c>b && c>d)

    {

        return c;

    }

    else

    {

        return d;

    }

}

int main() {

    int a, b, c, d;

    scanf("%d %d %d %d", &a, &b, &c, &d);

    int ans = max\_of\_four(a, b, c, d);

    printf("%d", ans);

    return 0;

}

Rectangle area

#include <iostream>

using namespace std;

/\*

 \* Create classes Rectangle and RectangleArea

 \*/

class Rectangle

{

    public:

    int width,height;

    void display()

    {

        cout<<width<<" "<<height<<endl;

    }

};

class RectangleArea : public Rectangle

{

    public:

    void read\_input()

    {

        cin>>width>>height;

    }

    void display()

    {

        cout<<width\*height;

    }

};

int main()

{

    /\*

     \* Declare a RectangleArea object

     \*/

    RectangleArea r\_area;

    /\*

     \* Read the width and height

     \*/

    r\_area.read\_input();

    /\*

     \* Print the width and height

     \*/

    r\_area.Rectangle::display();

    /\*

     \* Print the area

     \*/

    r\_area.display();

    return 0;

}

Pointer

#include <stdio.h>

void update(int \*a,int \*b) {

    // Complete this function

        int temp=\*a;

        \*a=\*a+\*b;

        if(temp>\*b)

        {

            \*b=temp-\*b;

        }

        else

        {

            \*b=\*b-temp;

        }

}

int main() {

    int a, b;

    int \*pa = &a, \*pb = &b;

    scanf("%d %d", &a, &b);

    update(pa, pb);

    printf("%d\n%d", a, b);

    return 0;

}

Array introduction

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

    /\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

    int n;

    cin>>n;

    int a[n];

    for(int i=0; i<n; i++)

    {

        cin>>a[i];

    }

    for(int i=n-1; i>=0; i--)

       {

          printf("%d ",a[i]);

       }

    return 0;

}

Strings

#include <iostream>

#include <string>

using namespace std;

int main() {

    // Complete the program

    string a;

    string b;

    cin>>a>>b;

    cout<<a.size()<<" "<<b.size()<<endl;

    cout<<a+b<<endl;

    char c1 =a.at(0);

    char c2 =b.at(0);

    string s1;

    string s2;

    if (!a.empty()) {

        s1=a.substr(1, a.size() - 1);

    }

    if (!b.empty()) {

        s2=b.substr(1, b.size() - 1);

    }

    cout<<c2+s1<<" "<<c1+s2<<endl;

    return 0;

}

Inheritance rectangle area

#include <iostream>

using namespace std;

/\*

 \* Create classes Rectangle and RectangleArea

 \*/

class Rectangle

{

    public:

    int width,height;

    void display()

    {

        cout<<width<<" "<<height<<endl;

    }

};

class RectangleArea : public Rectangle

{

    public:

    void read\_input()

    {

        cin>>width>>height;

    }

    void display()

    {

        cout<<width\*height;

    }

};

int main()

{

    /\*

     \* Declare a RectangleArea object

     \*/

    RectangleArea r\_area;

    /\*

     \* Read the width and height

     \*/

    r\_area.read\_input();

    /\*

     \* Print the width and height

     \*/

    r\_area.Rectangle::display();

    /\*

     \* Print the area

     \*/

    r\_area.display();

    return 0;

}

Class

#include <iostream>

#include <sstream>

using namespace std;

/\*

Enter code for class Student here.

Read statement for specification.

\*/

class Student

{

    private:

    int ag,stand;

    string fname, lname;

    public:

    void set\_age(int age)

    {

        ag=age;

    }

    int get\_age()

    {

        return ag;

    }

    void set\_first\_name(string first\_name)

    {

        fname=first\_name;

    }

    string get\_first\_name()

    {

        return fname;

    }

    void set\_last\_name(string last\_name)

    {

        lname=last\_name;

    }

    string get\_last\_name()

    {

        return lname;

    }

    void set\_standard(int standard)

    {

        stand=standard;

    }

    int get\_standard()

    {

        return stand;

    }

    void to\_string()

    {

        cout<<ag<<","<<fname<<","<<lname<<","<<stand;

    }

};

int main() {

    int age, standard;

    string first\_name, last\_name;

    cin >> age >> first\_name >> last\_name >> standard;

    Student st;

    st.set\_age(age);

    st.set\_standard(standard);

    st.set\_first\_name(first\_name);

    st.set\_last\_name(last\_name);

    cout << st.get\_age() << "\n";

    cout << st.get\_last\_name() << ", " << st.get\_first\_name() << "\n";

    cout << st.get\_standard() << "\n";

    cout << "\n";

    st.to\_string();

    return 0;

}

Inheritance introduction

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class Triangle{

    public:

        void triangle(){

            cout<<"I am a triangle\n";

        }

};

class Isosceles : public Triangle{

    public:

        void isosceles(){

            cout<<"I am an isosceles triangle\n";

        }

        //Write your code here.

          void description()

          {

              cout<<"In an isosceles triangle two sides are equal"<<endl;

          }

};

int main(){

    Isosceles isc;

    isc.isosceles();

    isc.description();

    isc.triangle();

    return 0;

}

Multi level inheritance

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class Triangle{

    public:

        void triangle(){

            cout<<"I am a triangle\n";

        }

};

class Isosceles : public Triangle{

    public:

        void isosceles(){

            cout<<"I am an isosceles triangle\n";

        }

};

//Write your code here.

class Equilateral : public Isosceles

{

    public:

    void equilateral()

    {

        cout<<"I am an equilateral triangle"<<endl;

    }

};

int main(){

    Equilateral eqr;

    eqr.equilateral();

    eqr.isosceles();

    eqr.triangle();

    return 0;

}