Introduction & C++ Basics

LI Hao 李颢, Assoc. Prof. SPEIT & Dept. Automation of SEIEE

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

General Info

- **Evaluation**
  - Class performance: 10%
  - Online quiz: 20% (4 x 5%)
  - Course project: 30% (2 x 15%)
  - Written exam: 40%
- **Teaching assistant**
  - Zhuoye YING应卓冶
  - E-mail: 835219559@qq.com
- **Reference**
  - C. Shaffer, "A Practical Introduction to Data Structures and Algorithm Analysis", 3rd, 2011

群聊：2023数据结构ICE3402P【李】

- Real names in the wechat group
- Resign if not registered

Data Structures & Algorithms

- **Data Structures**
  - Reasonable representation and organization of data so that they can be manipulated in a reasonable way (by certain algorithm)
- **Algorithms**
  - Reasonable manipulation of data that are represented and organized in a reasonable way (as certain data structure)

Data Structures & Algorithms

- **Data Structures**
  - Reasonable representation and organization of data so that they can be manipulated in a reasonable way (by certain algorithm)
- **Algorithms**
  - Reasonable manipulation of data that are represented and organized in a reasonable way (as certain data structure)

Looking up characters

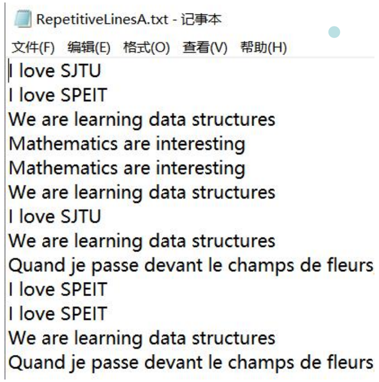
⇕

Alphabetically-organized characters

Data Structures & Algorithms

Demo I : Remove repetitive lines

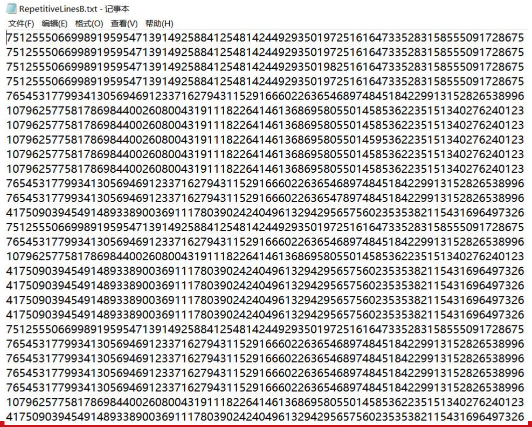
- Given data stored line by line in a file, remove repetitive lines of the file



RepetitiveLinesA.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

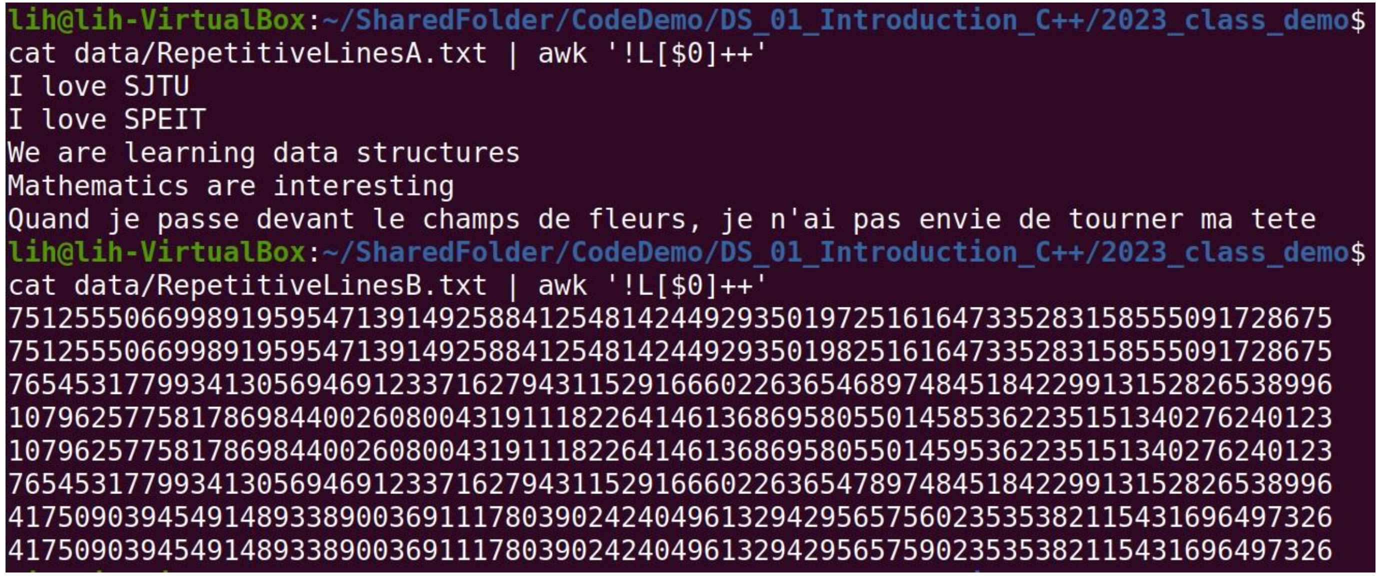
```
I love SJTU
I love SPEIT
We are learning data structures
Mathematics are interesting
Mathematics are interesting
We are learning data structures
I love SJTU
We are learning data structures
Quand je passe devant le champs de fleurs
I love SPEIT
I love SPEIT
We are learning data structures
Quand je passe devant le champs de fleurs
```



RepetitiveLinesB.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

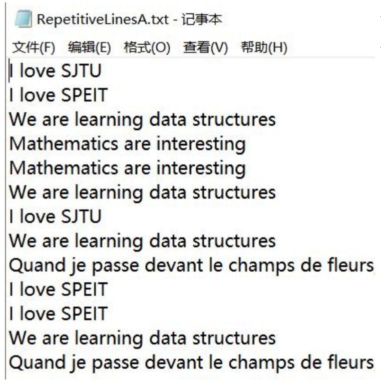
```
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019825161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014595362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365478974845184229913152826538996
41750903945491489338900369111780390242404961329429565756023535382115431696497326
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
41750903945491489338900369111780390242404961329429565756023535382115431696497326
```



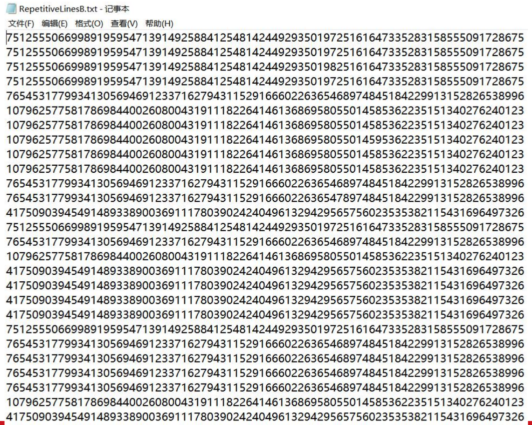
```
lih@lih-VirtualBox:~/SharedFolder/CodeDemo/DS_01_Introduction_C++/2023_class_demo$
cat data/RepetitiveLinesA.txt | awk '!L[$0]++'
I love SJTU
I love SPEIT
We are learning data structures
Mathematics are interesting
Quand je passe devant le champs de fleurs, je n'ai pas envie de tourner ma tete
lih@lih-VirtualBox:~/SharedFolder/CodeDemo/DS_01_Introduction_C++/2023_class_demo$
cat data/RepetitiveLinesB.txt | awk '!L[$0]++'
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019825161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014595362235151340276240123
76545317799341305694691233716279431152916660226365478974845184229913152826538996
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565759023535382115431696497326
```

Demo II: Search for a pattern

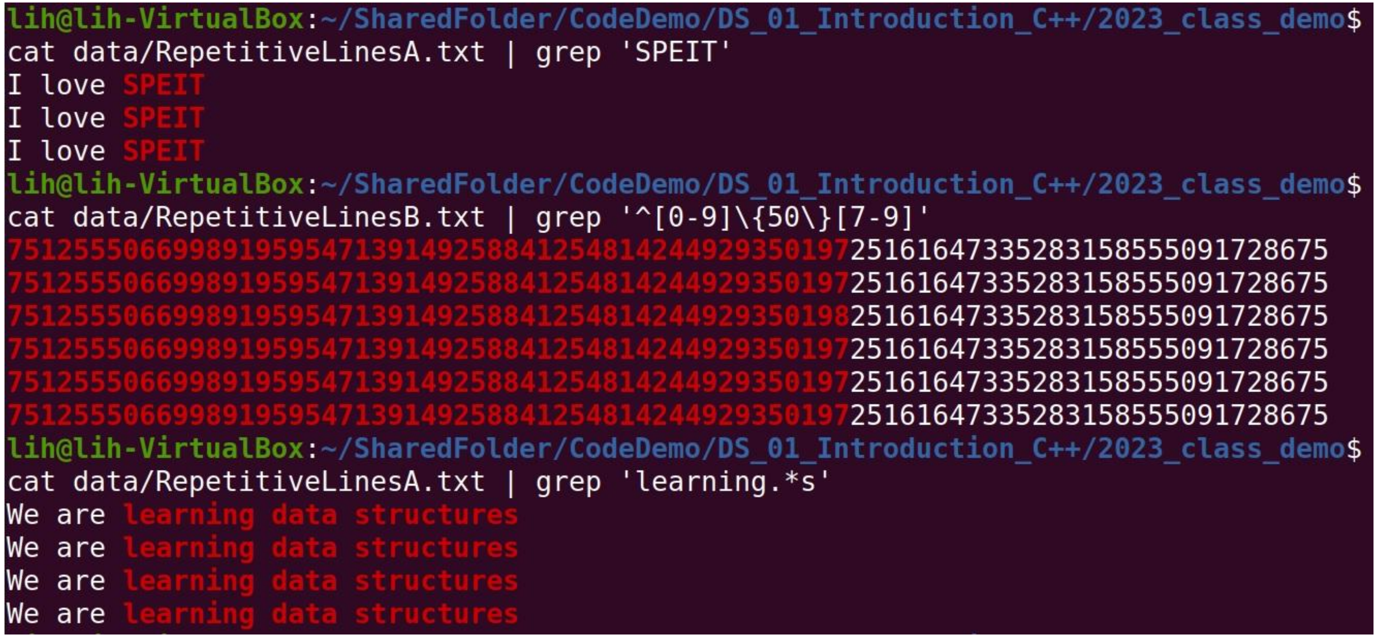
– Search for a specified phrase pattern in a file


RepetitiveLinesA.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
I love SJTU
I love SPEIT
We are learning data structures
Mathematics are interesting
Mathematics are interesting
We are learning data structures
I love SJTU
We are learning data structures
Quand je passe devant le champs de fleurs
I love SPEIT
I love SPEIT
We are learning data structures
Quand je passe devant le champs de fleurs
```


RepetitiveLinesB.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019825161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014595362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
10796257758178698440026080043191118226414613686958055014585362235151340276240123
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365478974845184229913152826538996
41750903945491489338900369111780390242404961329429565756023535382115431696497326
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
41750903945491489338900369111780390242404961329429565756023535382115431696497326
75125550669989195954713914925884125481424492935019725161647335283158555091728675
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
76545317799341305694691233716279431152916660226365468974845184229913152826538996
10796257758178698440026080043191118226414613686958055014585362235151340276240123
41750903945491489338900369111780390242404961329429565756023535382115431696497326
```



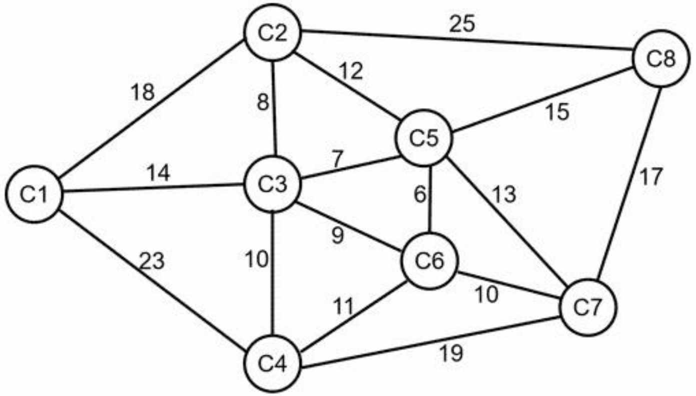
```
lih@lih-VirtualBox:~/SharedFolder/CodeDemo/DS_01_Introduction_C++/2023_class_demo$
cat data/RepetitiveLinesA.txt | grep 'SPEIT'
I love SPEIT
I love SPEIT
I love SPEIT
lih@lih-VirtualBox:~/SharedFolder/CodeDemo/DS_01_Introduction_C++/2023_class_demo$
cat data/RepetitiveLinesB.txt | grep '^[0-9]\{50\}[7-9]'
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019825161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
75125550669989195954713914925884125481424492935019725161647335283158555091728675
lih@lih-VirtualBox:~/SharedFolder/CodeDemo/DS_01_Introduction_C++/2023_class_demo$
cat data/RepetitiveLinesA.txt | grep 'learning.*s'
We are learning data structures
We are learning data structures
We are learning data structures
We are learning data structures
```

Data Structures & Algorithms

- **Demo III: Search for the optimal route**
  - Search for a route with the minimum distance from SPEIT to SEIEE

Heap priority queue

Minimum heap tree



C to C++: Object-Oriented Programming

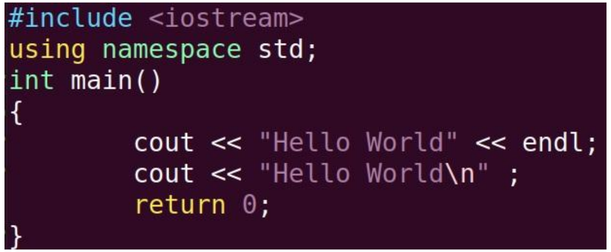
Bjarne Stroustrup
1983, designer of C++

- **C++ Merits**
  - Popular programming language
  - Compatible with C and computational efficiency
  - Embodiment of object-oriented programming
  - Data abstraction for data structures
- **What C++ adds to C**
  - Streams: input/output, files
  - Inlining, bool type, default argument values
  - **Class**: abstract data type
  - **Encapsulation**: data & methods
  - **Information hiding**: public, private, protected
  - **Polymorphism**: overloading, templates, virtual functions
  - **Inheritance**: derived classes, public interfaces

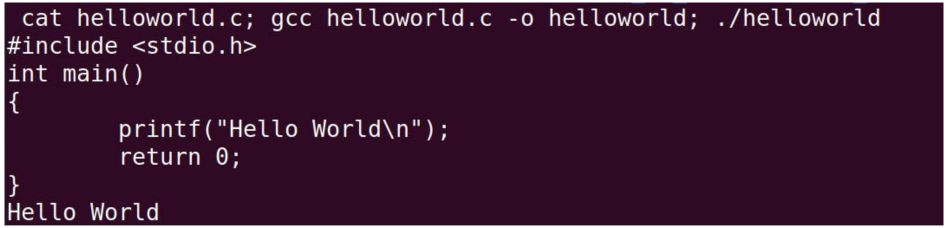
$\left.\right\}$ **Object-Oriented Programming**

Streams: Input/Output

- **helloworld.cpp**
  - Include file: **<iostream>**
  - File extensions convention for C++: ***.h**, ***.cpp**
  - Namespace: the most important is **std**

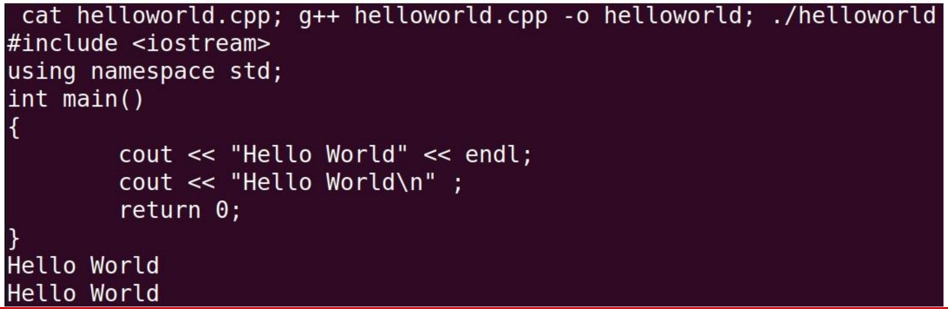


```
#include <iostream>
using namespace std;
int main()
{
        cout << "Hello World" << endl;
        cout << "Hello World\n" ;
        return 0;
}
```



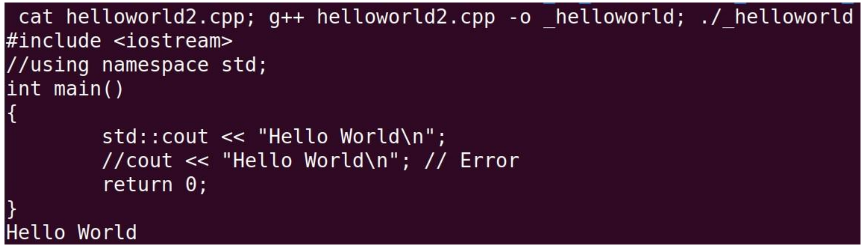
```
 cat helloworld.c; gcc helloworld.c -o helloworld; ./helloworld
#include <stdio.h>
int main()
{
        printf("Hello World\n");
        return 0;
}
Hello World
```

C



```
 cat helloworld.cpp; g++ helloworld.cpp -o helloworld; ./helloworld
#include <iostream>
using namespace std;
int main()
{
        cout << "Hello World" << endl;
        cout << "Hello World\n" ;
        return 0;
}
Hello World
Hello World
```

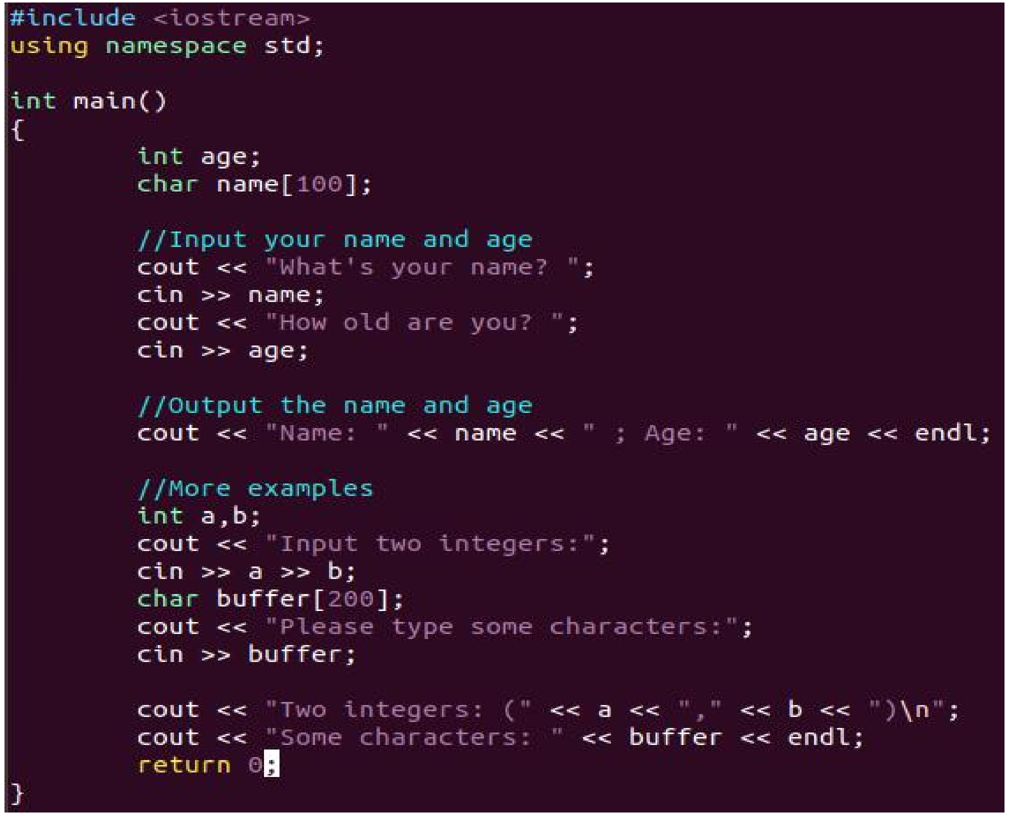
C++



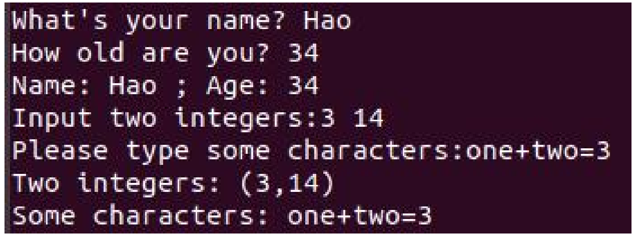
```
 cat helloworld2.cpp; g++ helloworld2.cpp -o _helloworld; ./_helloworld
#include <iostream>
//using namespace std;
int main()
{
        std::cout << "Hello World\n";
        //cout << "Hello World\n"; // Error
        return 0;
}
Hello World
```

Streams: Input/Output

- **Read & Write**
  - Read: **cin>>**
  - Write: **cout<<**



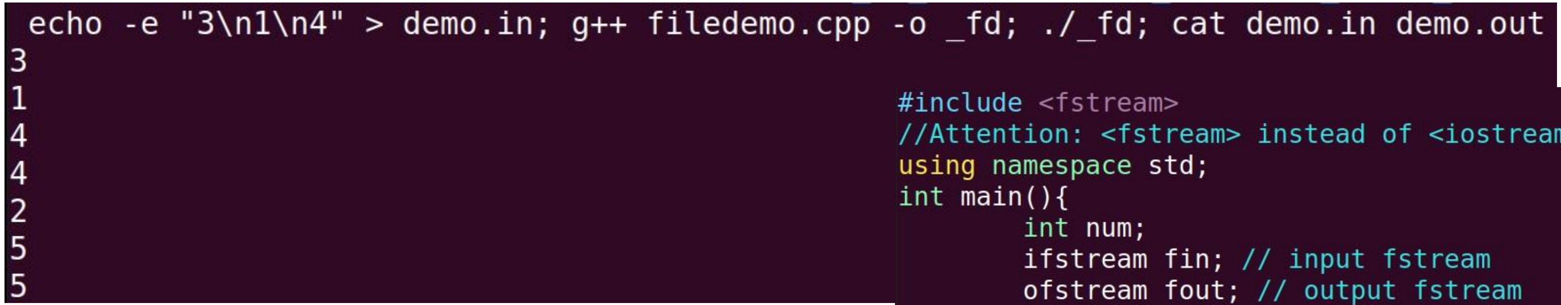
```cpp
#include <iostream>
using namespace std;

int main()
{
        int age;
        char name[100];

        //Input your name and age
        cout << "What's your name? ";
        cin >> name;
        cout << "How old are you? ";
        cin >> age;

        //Output the name and age
        cout << "Name: " << name << " ; Age: " << age << endl;

        //More examples
        int a,b;
        cout << "Input two integers:";
        cin >> a >> b;
        char buffer[200];
        cout << "Please type some characters:";
        cin >> buffer;

        cout << "Two integers: (" << a << "," << b << ")\n";
        cout << "Some characters: " << buffer << endl;
        return 0;
}
```



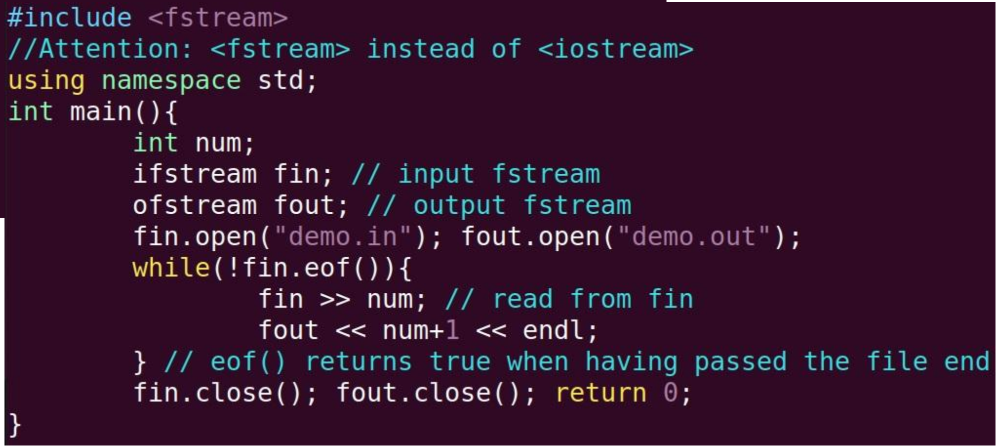
```
What's your name? Hao
How old are you? 34
Name: Hao ; Age: 34
Input two integers:3 14
Please type some characters:one+two=3
Two integers: (3,14)
Some characters: one+two=3
```

Streams: Files

- **Read & Write**
  - Include file: **<fstream>**
  - Read from a file: **ifstream fin>>**
  - Write into a file: **ofstream fout<<**



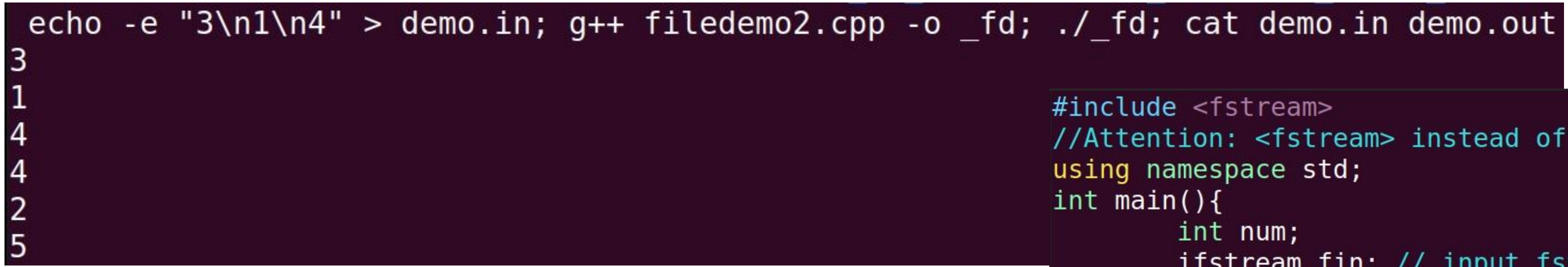
```
 echo -e "3\n1\n4" > demo.in; g++ filedemo.cpp -o _fd; ./_fd; cat demo.in demo.out
3
1
4
4
2
5
5
```



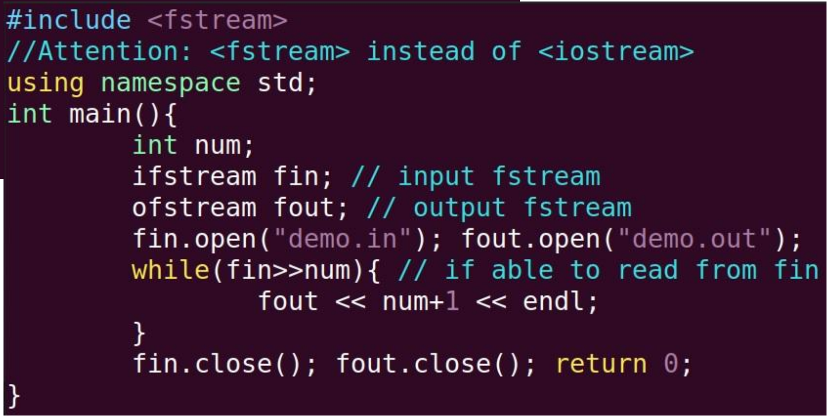
```
#include <fstream>
//Attention: <fstream> instead of <iostream>
using namespace std;
int main(){
        int num;
        ifstream fin; // input fstream
        ofstream fout; // output fstream
        fin.open("demo.in"); fout.open("demo.out");
        while(!fin.eof()){
                fin >> num; // read from fin
                fout << num+1 << endl;
        } // eof() returns true when having passed the file end
        fin.close(); fout.close(); return 0;
}
```

Streams: Files

- **Read & Write**
  - Include file: **<fstream>**
  - Read from a file: **ifstream fin>>**
  - Write into a file: **ofstream fout<<**



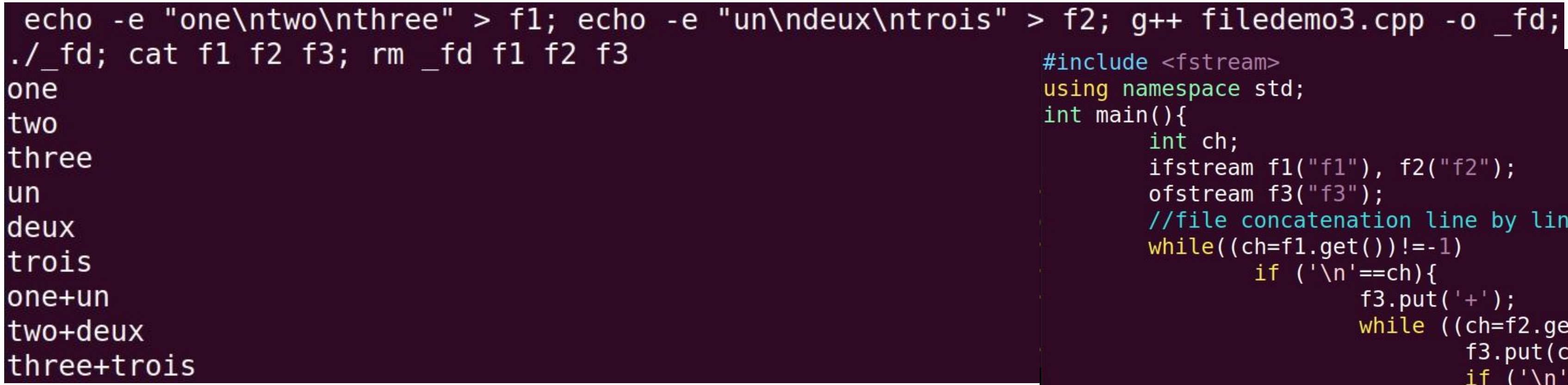
```
 echo -e "3\n1\n4" > demo.in; g++ filedemo2.cpp -o _fd; ./_fd; cat demo.in demo.out
3
1
4
4
2
5
```



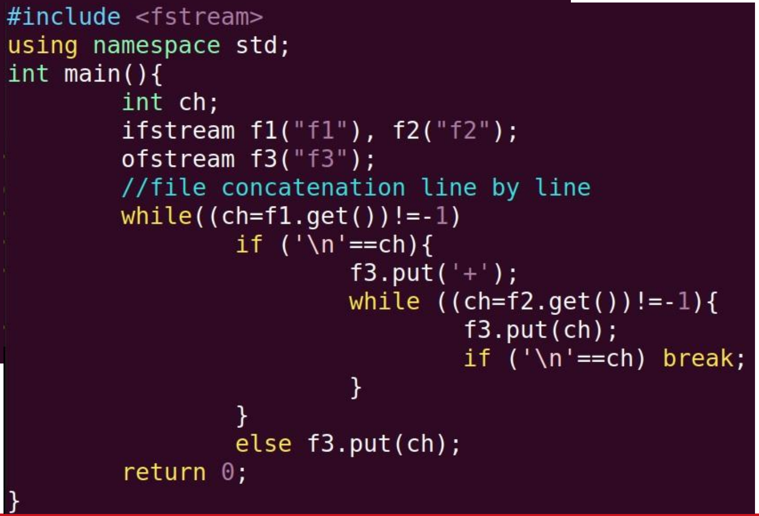
```
#include <fstream>
//Attention: <fstream> instead of <iostream>
using namespace std;
int main(){
        int num;
        ifstream fin; // input fstream
        ofstream fout; // output fstream
        fin.open("demo.in"); fout.open("demo.out");
        while(fin>>num){ // if able to read from fin
                fout << num+1 << endl;
        }
        fin.close(); fout.close(); return 0;
}
```

Streams: Files

- **Read & Write**
  - Include file: **<fstream>**
  - Read from a file: **ifstream fin>>**
  - Write into a file: **ofstream fout<<**



```
 echo -e "one\ntwo\nthree" > f1; echo -e "un\ndeux\ntrois" > f2; g++ filedemo3.cpp -o _fd;
./_fd; cat f1 f2 f3; rm _fd f1 f2 f3
one
two
three
un
deux
trois
one+un
two+deux
three+trois
```



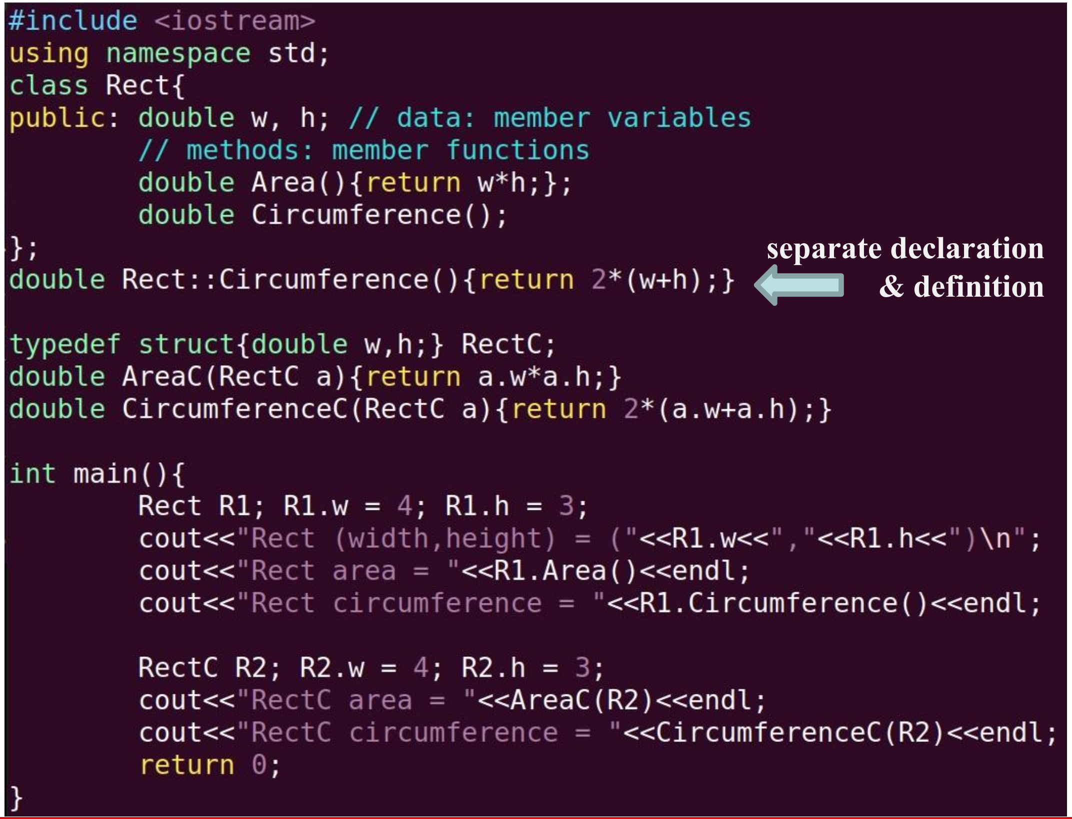
```
#include <fstream>
using namespace std;
int main(){
        int ch;
        ifstream f1("f1"), f2("f2");
        ofstream f3("f3");
        //file concatenation line by line
        while((ch=f1.get())!=-1)
                if ('\n'==ch){
                        f3.put('+');
                        while ((ch=f2.get())!=-1){
                                f3.put(ch);
                                if ('\n'==ch) break;
                        }
                }
                else f3.put(ch);
        return 0;
}
```

Class: Object-Oriented Programming

- **Encapsulation**

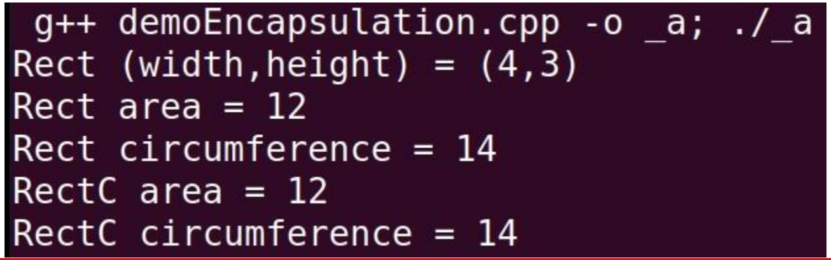
C++
- Data
- **Methods**

C
- Data only



```
#include <iostream>
using namespace std;
class Rect{
public: double w, h; // data: member variables
        // methods: member functions
        double Area(){return w*h;};
        double Circumference();
};
double Rect::Circumference(){return 2*(w+h);}

typedef struct{double w,h;} RectC;
double AreaC(RectC a){return a.w*a.h;}
double CircumferenceC(RectC a){return 2*(a.w+a.h);}

int main(){
        Rect R1; R1.w = 4; R1.h = 3;
        cout<<"Rect (width,height) = ("<<R1.w<<","<<R1.h<<")\n";
        cout<<"Rect area = "<<R1.Area()<<endl;
        cout<<"Rect circumference = "<<R1.Circumference()<<endl;

        RectC R2; R2.w = 4; R2.h = 3;
        cout<<"RectC area = "<<AreaC(R2)<<endl;
        cout<<"RectC circumference = "<<CircumferenceC(R2)<<endl;
        return 0;
}
```

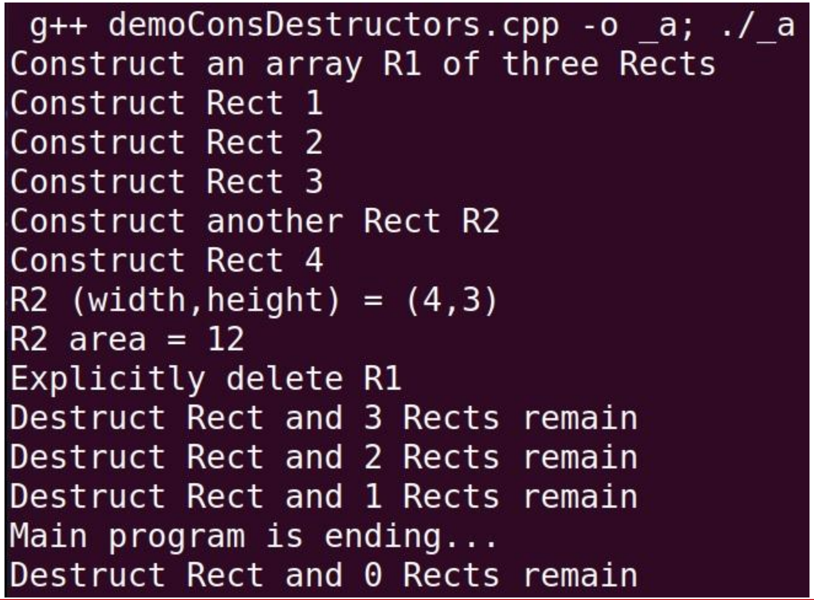
separate declaration & definition



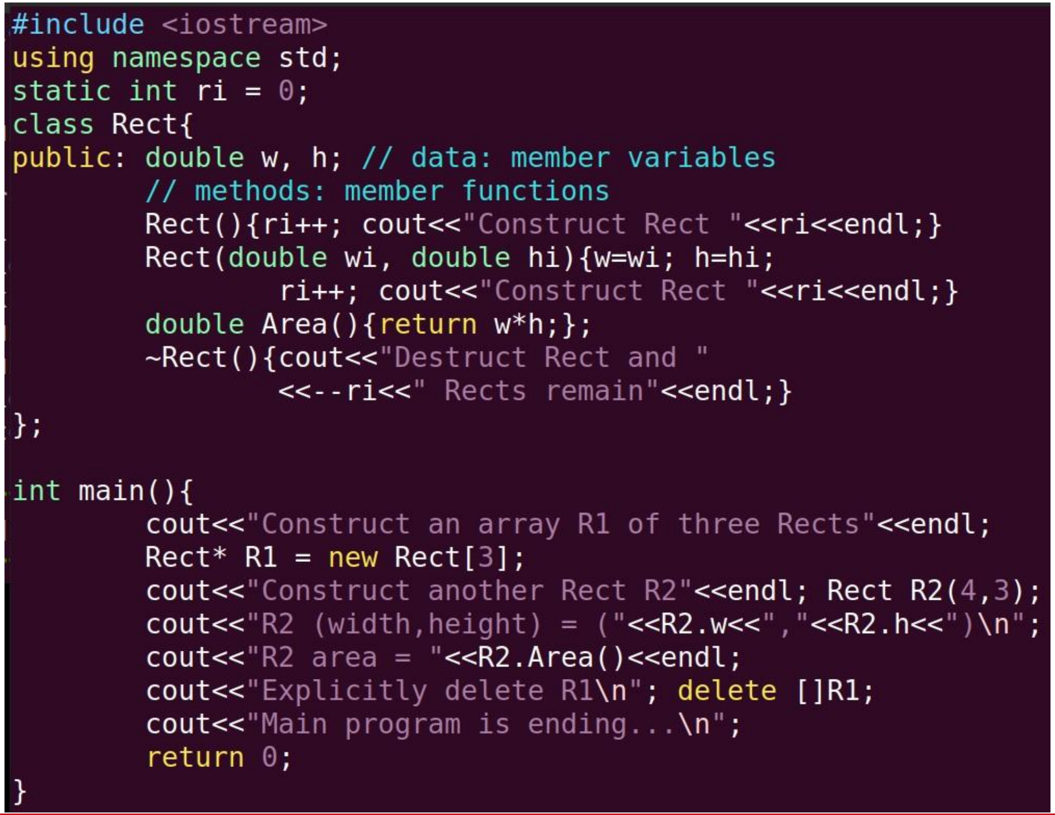
```
 g++ demoEncapsulation.cpp -o _a; ./_a
Rect (width,height) = (4,3)
Rect area = 12
Rect circumference = 14
RectC area = 12
RectC circumference = 14
```

Class: Object-Oriented Programming

- **Encapsulation**
  - Constructors
  - Destructors



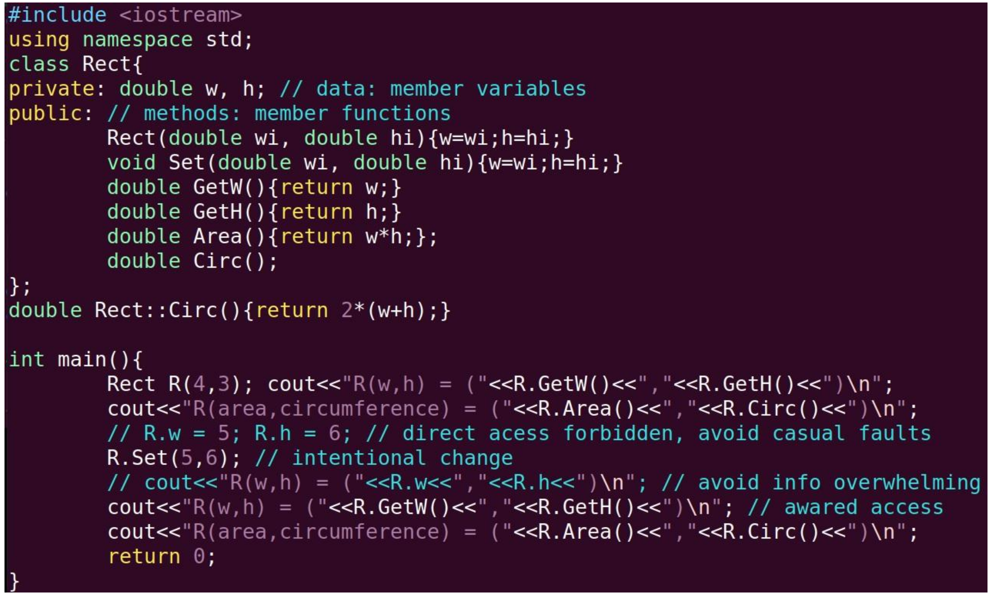
```
 g++ demoConsDestructors.cpp -o _a; ./_a
Construct an array R1 of three Rects
Construct Rect 1
Construct Rect 2
Construct Rect 3
Construct another Rect R2
Construct Rect 4
R2 (width,height) = (4,3)
R2 area = 12
Explicitly delete R1
Destruct Rect and 3 Rects remain
Destruct Rect and 2 Rects remain
Destruct Rect and 1 Rects remain
Main program is ending...
Destruct Rect and 0 Rects remain
```



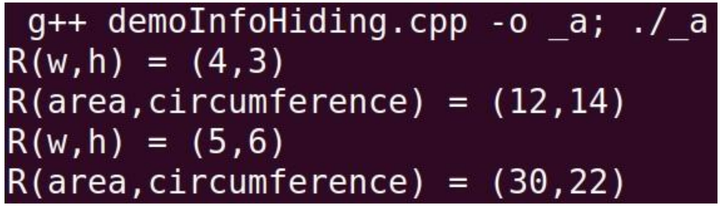
```cpp
#include <iostream>
using namespace std;
static int ri = 0;
class Rect{
public: double w, h; // data: member variables
        // methods: member functions
        Rect(){ri++; cout<<"Construct Rect "<<ri<<endl;}
        Rect(double wi, double hi){w=wi; h=hi;
                ri++; cout<<"Construct Rect "<<ri<<endl;}
        double Area(){return w*h;};
        ~Rect(){cout<<"Destruct Rect and "
                <<--ri<<" Rects remain"<<endl;}
};

int main(){
        cout<<"Construct an array R1 of three Rects"<<endl;
        Rect* R1 = new Rect[3];
        cout<<"Construct another Rect R2"<<endl; Rect R2(4,3);
        cout<<"R2 (width,height) = ("<<R2.w<<","<<R2.h<<")\n";
        cout<<"R2 area = "<<R2.Area()<<endl;
        cout<<"Explicitly delete R1\n"; delete []R1;
        cout<<"Main program is ending...\n";
        return 0;
}
```

Class: Object-Oriented Programming

- **Information hiding**
  - public, **private**, protected
  - Avoid casual faults or unawared altering
  - Avoid info overwhelming
  - Black-box methodology



```cpp
#include <iostream>
using namespace std;
class Rect{
private: double w, h; // data: member variables
public: // methods: member functions
        Rect(double wi, double hi){w=wi;h=hi;}
        void Set(double wi, double hi){w=wi;h=hi;}
        double GetW(){return w;}
        double GetH(){return h;}
        double Area(){return w*h;};
        double Circ();
};
double Rect::Circ(){return 2*(w+h);}

int main(){
        Rect R(4,3); cout<<"R(w,h) = ("<<R.GetW()<<","<<R.GetH()<<")\n";
        cout<<"R(area,circumference) = ("<<R.Area()<<","<<R.Circ()<<")\n";
        // R.w = 5; R.h = 6; // direct acess forbidden, avoid casual faults
        R.Set(5,6); // intentional change
        // cout<<"R(w,h) = ("<<R.w<<","<<R.h<<")\n"; // avoid info overwhelming
        cout<<"R(w,h) = ("<<R.GetW()<<","<<R.GetH()<<")\n"; // awared access
        cout<<"R(area,circumference) = ("<<R.Area()<<","<<R.Circ()<<")\n";
        return 0;
}
```



```
 g++ demoInfoHiding.cpp -o _a; ./_a
R(w,h) = (4,3)
R(area,circumference) = (12,14)
R(w,h) = (5,6)
R(area,circumference) = (30,22)
```

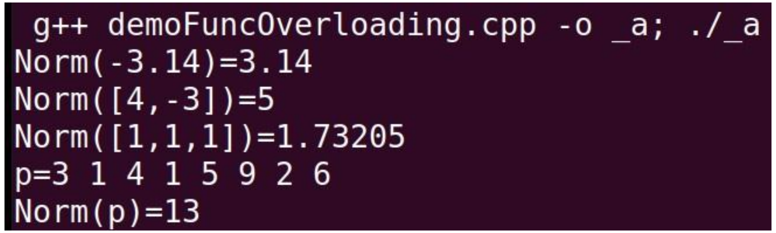
Class: Object-Oriented Programming

- **Information hiding**
  - public, **private**, protected
  - Avoid casual faults or unawared altering
  - Avoid info overwhelming
  - Black-box methodology

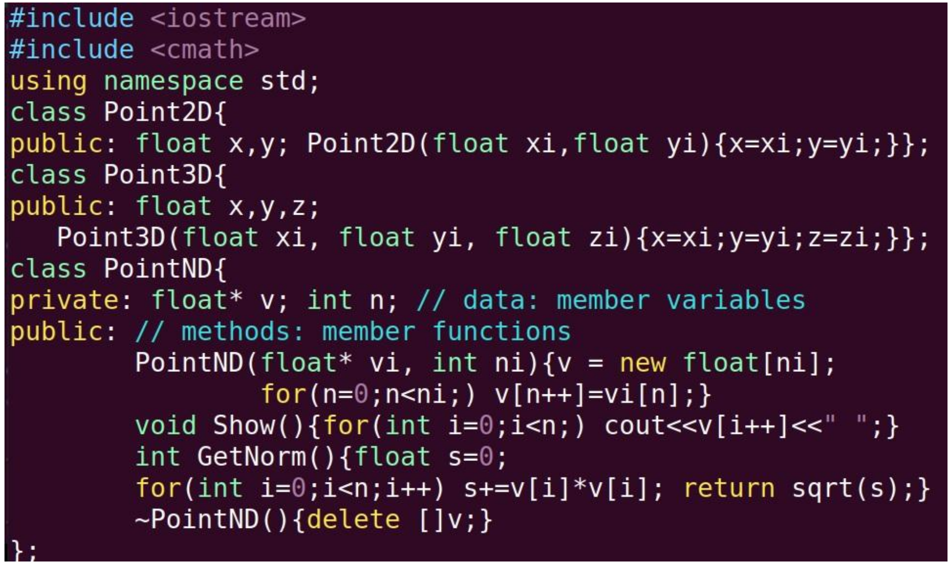
private	*class*'s members & friends can use
protected	*class*'s members & friends and *derived classes*'s members & friends can use
public	general public can use

Class: Object-Oriented Programming

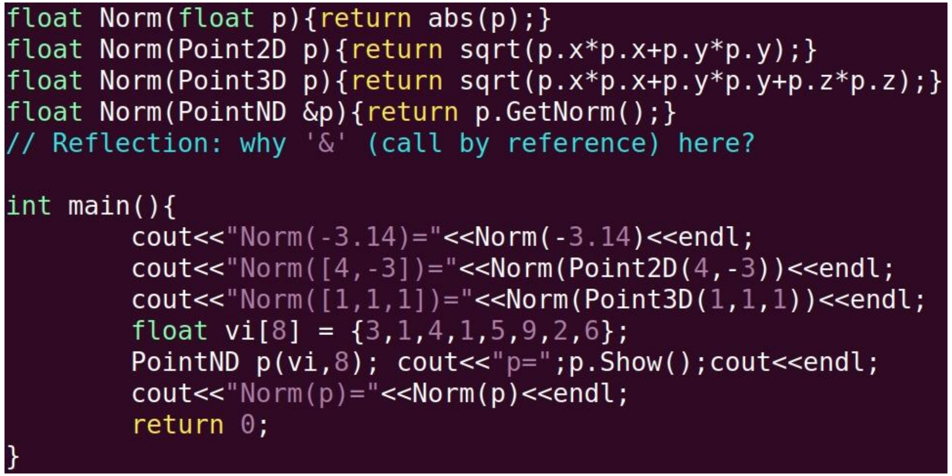
- **Polymorphism**
  - **Function overloading**
  - Operator overloading



```
 g++ demoFuncOverloading.cpp -o _a; ./_a
Norm(-3.14)=3.14
Norm([4,-3])=5
Norm([1,1,1])=1.73205
p=3 1 4 1 5 9 2 6
Norm(p)=13
```



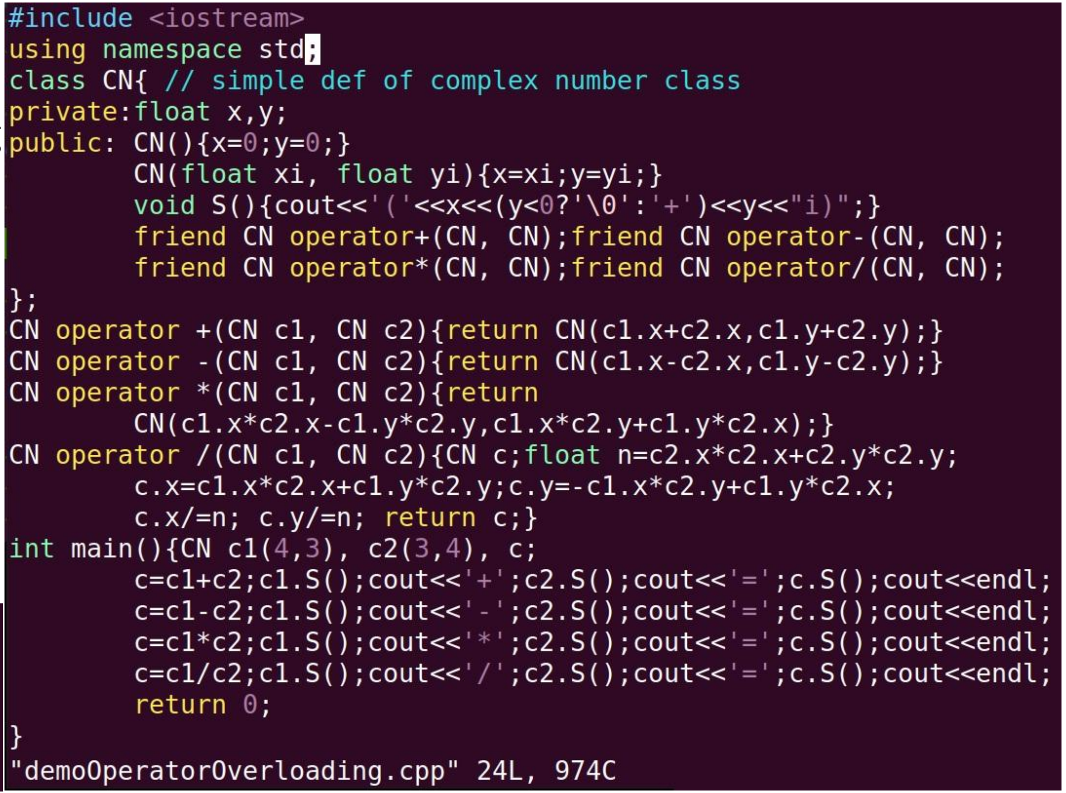
```
#include <iostream>
#include <cmath>
using namespace std;
class Point2D{
public: float x,y; Point2D(float xi,float yi){x=xi;y=yi;}};
class Point3D{
public: float x,y,z;
   Point3D(float xi, float yi, float zi){x=xi;y=yi;z=zi;}};
class PointND{
private: float* v; int n; // data: member variables
public: // methods: member functions
        PointND(float* vi, int ni){v = new float[ni];
                for(n=0;n<ni;) v[n++]=vi[n];}
        void Show(){for(int i=0;i<n;) cout<<v[i++]<<" ";}
        int GetNorm(){float s=0;
        for(int i=0;i<n;i++) s+=v[i]*v[i]; return sqrt(s);}
        ~PointND(){delete []v;}
};
```



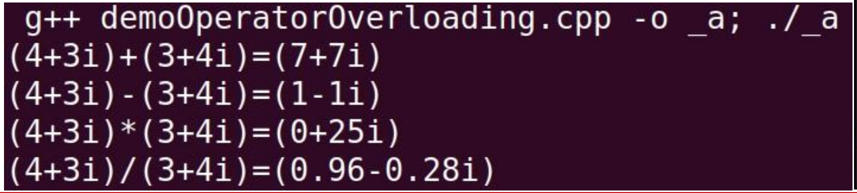
```
float Norm(float p){return abs(p);}
float Norm(Point2D p){return sqrt(p.x*p.x+p.y*p.y);}
float Norm(Point3D p){return sqrt(p.x*p.x+p.y*p.y+p.z*p.z);}
float Norm(PointND &p){return p.GetNorm();}
// Reflection: why '&' (call by reference) here?

int main(){
        cout<<"Norm(-3.14)="<<Norm(-3.14)<<endl;
        cout<<"Norm([4,-3])="<<Norm(Point2D(4,-3))<<endl;
        cout<<"Norm([1,1,1])="<<Norm(Point3D(1,1,1))<<endl;
        float vi[8] = {3,1,4,1,5,9,2,6};
        PointND p(vi,8); cout<<"p=";p.Show();cout<<endl;
        cout<<"Norm(p)="<<Norm(p)<<endl;
        return 0;
}
```

Class: Object-Oriented Programming

- **Polymorphism**
  - Function overloading
  - **Operator overloading**



```
#include <iostream>
using namespace std;
class CN{ // simple def of complex number class
private:float x,y;
public: CN(){x=0;y=0;}
        CN(float xi, float yi){x=xi;y=yi;}
        void S(){cout<<'('<<x<<(y<0?'\0':'+')<<y<<"i)";}
        friend CN operator+(CN, CN);friend CN operator-(CN, CN);
        friend CN operator*(CN, CN);friend CN operator/(CN, CN);
};
CN operator +(CN c1, CN c2){return CN(c1.x+c2.x,c1.y+c2.y);}
CN operator -(CN c1, CN c2){return CN(c1.x-c2.x,c1.y-c2.y);}
CN operator *(CN c1, CN c2){return
        CN(c1.x*c2.x-c1.y*c2.y,c1.x*c2.y+c1.y*c2.x);}
CN operator /(CN c1, CN c2){CN c;float n=c2.x*c2.x+c2.y*c2.y;
        c.x=c1.x*c2.x+c1.y*c2.y;c.y=-c1.x*c2.y+c1.y*c2.x;
        c.x/=n; c.y/=n; return c;}
int main(){CN c1(4,3), c2(3,4), c;
        c=c1+c2;c1.S();cout<<'+';c2.S();cout<<'=';c.S();cout<<endl;
        c=c1-c2;c1.S();cout<<'-';c2.S();cout<<'=';c.S();cout<<endl;
        c=c1*c2;c1.S();cout<<'*';c2.S();cout<<'=';c.S();cout<<endl;
        c=c1/c2;c1.S();cout<<'/';c2.S();cout<<'=';c.S();cout<<endl;
        return 0;
}
"demoOperatorOverloading.cpp" 24L, 974C
```



```
 g++ demoOperatorOverloading.cpp -o _a; ./_a
(4+3i)+(3+4i)=(7+7i)
(4+3i)-(3+4i)=(1-1i)
(4+3i)*(3+4i)=(0+25i)
(4+3i)/(3+4i)=(0.96-0.28i)
```

THANK YOU

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY