C++程序设计

主讲：王老师

尚德机构

学习是一种信仰

考试题型

单选题　　1分×20题 = 20分

填空题　　1分×15题 = 15分

程序填空题　4分×5题 = 20分

程序分析题　6分×5题 = 30分

程序设计题　　2题 = 15分

（第一题5分，第二题10分）

第八章　文件操作

本章主要内容

- 文件基本概念和文件流类
- 打开和关闭文件
- 文件读写操作
- 随机访问文件

8.1 文件基本概念和文件流类

8.1 文件基本概念和文件流类

- 从不同的角度来看待文件就可以得到不同的文件分类。C++根据文件数据的编码方式不同分为文本文件和二进制文件。根据存取方式不同分为顺序存取文件和随机存取文件。

- 所谓“文本文件”和“二进制文件”是从文件格式的角度进行分类，是约定俗成的、从计算机用户角度出发进行的分类。

8.1.1 文件的概念

- 所谓的“顺序存取文件”和“随机存取文件”是根据访问文件中数据的方式来划分的。顺序存取文件就是按照文件中数据存储次序进行顺序操作，为访问第i个数据，就首先要访问第i-1个数据，在整个文件操作过程中，将移动位置指针的工作交给系统自动完成。磁带文件就是一个典型的顺序存取文件。随机访问文件是根据应用的需要，通过命令移动位置指针直接定位到文件内需要的位置并进行数据操作。
- 对文件的基本操作分为读文件和写文件。所谓“读文件”就是将文件中的数据读入内存之中，也称为“输入”。所谓“写文件”就是将内存中的数据存入文件之中，也称为“输出”。

8.1.2 C++文件流类

C++标准类库中有3个流类可以用于文件操作，这3个类统称为文件流类，分别如下：

1)ifstream：用于从文件中读取数据。

2)ofstream：用于向文件中写入数据。

3)fstream：既可用于从文件中读取数据，又可用于向文件中写入数据。

图 7-1 iostream 流类库的类关系图

使用这3个流类时，程序中需要包含fstream头文件。

类ifstream和类fstream都是从类istream派生而来的，因此类ifstream拥有类istream的全部成员函数。同样，类ofstream和类fstream也拥有类ostream的全部成员函数。这3个类中有一些十分熟悉的成员函数可以使用，如operator<<、operator>>、peek( )、ignore( )、getline( )、get( )等。

在程序中，要使用一个文件，必须包含3个基本步骤：打开（open）文件——操作文件——关闭（close）文件。操作文件就是对文件进行读/写。

C++文件流类有相应的成员函数来实现打开、读、写、关闭等文件操作。

课堂练习

要进行文件的输出，除了包含头文件iostream外，还要包含头文件（　）

A:ifstream

B:fstream

C:ostream

D:cstdio

课堂练习

要进行文件的输出，除了包含头文件iostream外，还要包含头文件（　）

A:ifstream

B:fstream

C:ostream

D:cstdio

答案：B

8.2 打开和关闭文件

打开文件的方式有以下两种：

1)先建立流对象，然后调用open( )函数连接外部文件。格式如下：

流类名 对象名;

对象名.open(文件名,模式);

2)调用流类带参数的构造函数，在建立流对象的同时连接外部文件。格式如下：

流类名 对象名(文件名,模式);

其中的“流类”是C++流类库定义的文件流类ifstream、ofstream或fstream。若要以读方式打开文件则应使用类ifstream，若以写方式打开文件则应使用类ofstream，若以读/写方式打开文件则应使用类fstream。

8.2.1 打开文件

模式标记	适用对象	作用
ios::in	ifstream fstream	以读方式打开文件。如果文件不存在，则打开出错
ios::out	ofstream fstream	以写方式打开文件。如果文件不存在，则新建该文件；如果文件已经存在，则打开时清除原来的内容
ios::app	ofstream	以追加方式打开文件，用于在文件尾部添加数据。如果文件不存在，则新建该文件
ios::ate	ofstream	打开一个已有的文件，并将文件读指针指向文件末尾。如果文件不存在，则打开出错
ios::trunc	ofstream	删除文件现有内容。单独使用时与ios::out相同
ios::binary	ifstream ofstream fstream	以二进制方式打开文件。若不指定此模式，则以默认的文本模式打开文件
ios::in | ios::out	fstream	打开已存在的文件，既可读取其内容，也可向其写入数据。文件刚打开时，原有内容保持不变。如果文件不存在，则打开出错
ios::in | ios::out	ofstream	打开已存在的文件，可以向其写入数据。文件刚打开时，原有内容保持不变。如果文件不存在，则打开出错
ios::in | ios::out | ios::trunc	fstream	打开文件，既可读取其内容，也可向其写入数据。如果文件本来就存在，则打开时清除原来的内容；如果文件不存在，则新建该文件

8.2.1 打开文件

例如，要从当前文件夹中名为data.txt的文件中读取数据，可以使用如下语句打开文件。

```
ifstream inFile;                    //建立输入文件流对象
inFile.open("data.txt",ios::in);    //连接文件，指定打开模式
```

也可以使用第二种方式打开，语句如下：

```
ifstream inFile("data.txt",ios::in);
```

调用ifstream类带参数的构造函数，在建立流对象的同时，用参数形式连接外部文件并指定打开模式。

要以读方式打开文本文件，还可以使用如下语句：

```
ifstream inFile;                    //建立输入文件流对象
inFile.open("data.txt");            //没有指定打开模式，默认以in方式打开文本文件
```

再比如，要在c盘的c2019文件夹中打开（创建）一个名为newfile的二进制文件，用于保存程序产生的数据，可以使用如下语句打开文件：

```
ofstream outFile;                   //建立输入文件流对象
outFile.open("c:\\c2019\\newfile",ios::out | ios::binary);    //连接文件，指定打开模式
```

也可以使用如下语句打开文件：

```
ofstream outFile("c:\\c2019\\newfile",ios::out | ios::binary);
```

8.2.2 关闭文件

使用fstream中的成员函数close( )关闭文件。

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ifstream inFile(“c:\\tmp\\test.txt”,ios::in);    //声明对象inFile并调用构造函数
    if(inFile)
    {
        cout<<"成功打开文件:c:\\tmp\\test.txt\n";
        inFile.close();
    }
    else
        cout<<"打开文件失败: c:\\tmp\\test.txt\n";
    ofstream outFile("test1.txt",ios::out);       //声明对象outFile并调用构造函数
    if(!outFile)
        cout<<"errorl"<<endl;
    else  {  cout<<“成功打开文件: test1.txt\n”;       outFile.close();  }
    fstream outFile2(“tmp\\test2.txt”,ios::out|ios::in);   //声明对象outFile2并调用构造函数
    if(outFile2)
    {    cout<<“成功打开文件:tmp\\test2.txt\n”;    outFile2.close();              }
    else
       cout<<"error2"<<endl;
    return 0;
}
```

```
成功打开文件:c:\tmp\test.txt
成功打开文件: test1.txt
成功打开文件:tmp\test2.txt
```

课堂练习

要求打开文件"d:\file.dat"，可写入数据，正确的语句是（　）

A:ifstream infile("d:\file.dat",ios::in);

B:ifstream infile("d:\\file.dat",ios::in);

C:ofstream infile("d:\file.dat",ios::out);

D:fstream infile("d:\\file.dat",ios::in|ios::out);

课堂练习

要求打开文件"d:\file.dat"，可写入数据，正确的语句是（　）

A:ifstream infile("d:\file.dat",ios::in);

B:ifstream infile("d:\\file.dat",ios::in);

C:ofstream infile("d:\file.dat",ios::out);

D:fstream infile("d:\\file.dat",ios::in|ios::out);

答案：D

打开文件的方式的格式：流类名 对象名(文件名,模式);
（1）C++流类库定义的文件流类对应的打开文件方式有：
若要以读方式打开文件则应使用类ifstream，
若以写方式打开文件则应使用类ofstream，
若以读/写方式打开文件则应使用类fstream。
（2）模式：
ios::in ： 以读方式打开文件。
ios::out ： 以写方式打开文件。
ios::in | ios::out：既可读取其内容，也可向其写入数据。

课堂练习

在C++中打开一个文件的目的之一是建立关联，其中，建立关联的是指定的文件与一个（　）。

A:类

B:流

C:对象

D:结构

课堂练习

在C++中打开一个文件的目的之一是建立关联，其中，建立关联的是指定的文件与一个（　）。

A:类

B:流

C:对象

D:结构

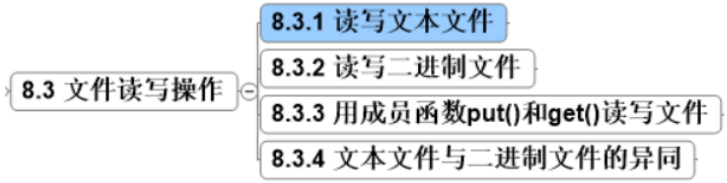
答案：B

打开文件有以下两个目的：

1)建立关联。通过指定文件名，建立起文件和文件流对象的关联，以后在对文件进行操作时，就可以通过与之关联的流对象来进行。故本题选B。

2)指明文件的使用方式和文件格式。

8.3 文件读写操作



假定现在要实现一个程序，从键盘输入学生的学号、姓名和成绩，将它们存入文件score.txt中。可以使用文本文件保存数据，文件中每一行保存一名学生的成绩信息，学生成绩信息的数据项之间通过空格符分隔，格式存储如下：

学号 姓名 成绩

为了方便程序实现，假设学号不超过10个字节，姓名不超过20个字节，成绩为整型，见程序8-3。

【程序8-3】对文本文件score.txt进行输入/输出

```
#include<iostream>
#include<fstream>
using namespace std;
int main( )
{
    char id[11],name[21];
    int score;
    ofstream outFile;
    outFile.open("score.txt",ios::out);   //以写方式打开文本文件
    if(!outFile)                          //条件成立，则说明文件打开出错
    {
        cout<<"创建文件失败"<<endl;
        return 0;
    }
    cout<<"请输入: 学号  姓名 成绩 (以Ctrl+Z结束输入)\n";
    while(cin>>id>>name>>score)
        outFile<<id<<" "<<name<<" "<<score<<endl; //向流中插入数据
    outFile.close();
    return 0;
}
```

```
请输入：学号 姓名 成绩（以 Ctrl+Z 结束输入）
2018001001 zhangsan 90↙
2018001002 lisi 9↙
2018001003 wangwu 85↙
2018001004 zhangsanfeng 100↙
2008001005 zhouwuzhengwangyishi 100↙
^Z↙
```

```
2018001001 zhangsan 90
2018001002 lisi 9
2018001003 wangwu 85
2018001004 zhangsanfeng 100
2008001005 zhouwuzhengwangyishi 100
```

```
学生学号      姓名                    成绩
2018001001 zhangsan                  90
2018001002 lisi                       9
2018001003 wangwu                    85
2018001004 zhangsanfeng             100
2008001005 zhouwuzhengwangyishi 100
```

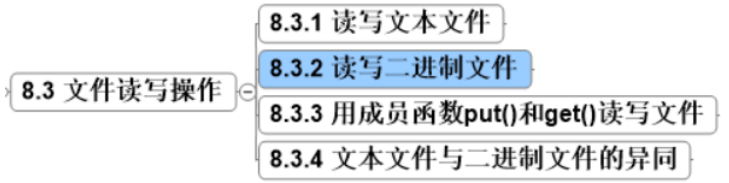
```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
int main( )
{
    char id[11],name[21];
    int score;
    ifstream inFile;
    inFile.open("score.txt",ios::in);       //以读方式打开文本文件
    if(!inFile)                             //条件成立，则说明文件打开出错
    {
        cout<<"打开文件失败"<<endl;
        return 0;
    }
    cout<<"学生学号  姓名\t\t\t成绩\n";
    while(inFile>>id>>name>>score)   //读入文件
        cout<<left<<setw(10)<<id<<" "<<setw(20)<<name<<" "<<setw(3)<<right<<score<<endl;//屏幕显示
    inFile.close( );
    return 0;
}
```

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
int main( )
{      char ch,filename[20];
       int count=0;                          //行号计数器
       bool newline=true;                    //开始一个新的标志
       cout<<"请输入文件名:";
       cin>>filename;
       ifstream inFile(filename,ios::in); //以读方式打开文本文件
       if(!inFile)                           //条件成立，则说明文件打开出错
       {     cout<<"打开文件失败"<<endl;     return 0;  }
       while( (ch=inFile.get())!=EOF)    //从流inFile中读入一个字符并判断
       {    if(newline)                      //若是新行开始，则显示行号
            {    cout<<setw(4)<<++count<<':';     newline=false;  //清除新行标志
            }
            if(ch=='\n')                     //若读入字符为'\n',则表示将开始一个新行
                 newline=true;               //设置新行标志
            cout<<ch;
       }
       inFile.close( );                      //关闭文件
       return 0;
}
```

运行程序8-5，输入文件名：score.txt，屏幕显示如下：

```
请输入文件名：score.txt
1: 2018001001 zhangsan 90
2: 2018001002 lisi 9
3: 2018001003 wangwu 85
4: 2018001004 zhangsanfeng 100
5: 2008001005 zhouwuzhengwangyishi 100
```

8.3.2 读写二进制文件



对二进制文件进行读写不能使用前面提到的类似于cin、cout从流中读写数据的方法。

C++用binary方式打开二进制文件，调用ifstream或fstream的read()成员函数从文件中读取数据，调用ofstream或fstream的write()成员函数向文件中写入数据。

1.用ostream::write()成员函数写文件

ofstream和fstream的write()成员函数继承自ostream类，原型如下：

ostream & write(char * buffer, int nCount);

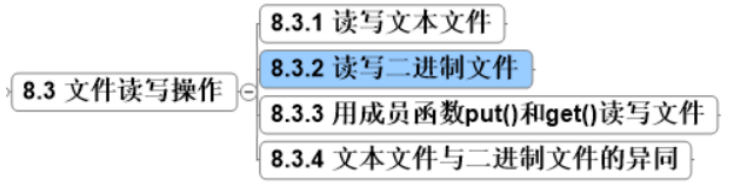
该成员函数将内存中buffer所指向的nCount个字节的内容写入文件，返回值是对函数所作用的对象的引用，如obj.write(...)的返回值就是对obj的引用。该函数是非格式化操作，将buffer所指的数据按字节序列直接存入文件中。

在使用write()与read()进行数据读写时，不必在数据之间再额外“插入”分隔符，这是因为它们都要求提供第2个参数来指定读写长度。

```
#include<iostream>
#include<fstream>
using namespace std;
class CStudent
{public:
        char id[11];
        char name[21];
        int score;
};
int main()
{
        CStudent stu;
        ofstream outFile("students.dat",ios::out|ios::binary); //以二进制写方式打开文本文件
        if(!outFile)                    //条件成立，则说明文件打开出错
        {
             cout<<"创建文件失败"<<endl;
            return 0;
        }
        cout<<"请输入: 学号  姓名 成绩 (以Ctrl+Z结束输入)\n";
        while(cin>>stu.id>>stu.name>>stu.score)
             outFile.write((char*)&stu,sizeof(stu)); //向文件中写入数据
        outFile.close( );                   //关闭文件
        return 0;
}
```

```
请输入：学号 姓名 成绩（以 Ctrl+Z 结束输入）
2019001001 ZhangSan 90↙
2019001002 LiSi 100↙
2019001003 WangWu 78↙
^Z↙
```

8.3.2 读写二进制文件



2.用istream::read( )成员函数读文件

ifstream和fstream的成员函数read()实际上继承自类istream，原型如下:

istream &read(char * buffer, int nCount);

该成员函数从文件中读取nCount个字节的内容，存放到buffer所指向的内存缓冲区中，返回值是对函数所作用的对象的引用。该函数是非格式化操作，对读取的字节序列不进行处理，直接存入buffer中，由程序的类型定义解释。

3.用ostream::gcount( )成员函数得到读取字节数

如果要知道每次读操作成功读取了多少个字节，可以在read()函数执行后立即调用文件流对象的成员函数gcount( )，其返回值就是最近一次read()函数执行时成功读取的字节数。

gcount()成员函数原型如下:

int gcount( );

8.3.3 用成员函数put( )和get( )读写文件

函数get( )有3种主要形式：

1.int get( );

不带参数的get( )函数从指定的输入流中提取一个字符（包含空白字符），函数的返回值即为该字符。当遇到文件结束符时，返回系统常量EOF。

2.istream& get(char &rch);

从指定输入流中提取一个字符（包含空白字符），将该字符作为rch引用的对象。当遇到文件结束符时，函数返回0；否则返回对istream对象的引用。

3.istream& get(char *pch, int nCount, char delim=’\n’);

从流的当前字符开始，读取nCount-1个字符，或遇到指定的分隔符delim结束。函数把读取的字符（不包括分隔符）写入数组pch中，并在字符串后添加结束符'\0'。

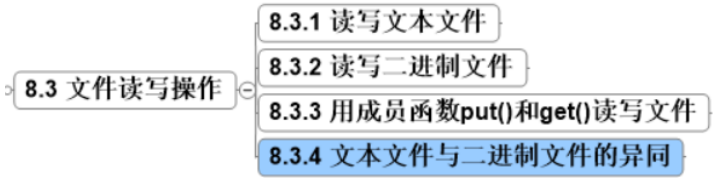
函数put()的语法格式如下：

ostream& put(char ch);

函数的功能是向输出流中插入一个字节。

成员函数get()和put()常用于读写字符或文本文件，但它们不仅仅可用于对字符的处理，而且对于二进制文件同样可以进行有效的处理。

8.3.4 文本文件与二进制文件的异同



- 在输入/输出过程中，系统要对内外存的数据格式进行相文本文件是以文本形式存储数据，其优点是具有较高的兼容性。缺点是存储一批纯数值信息时，要在数据之间人为地添加分隔符。应转换，文本文件的另一个缺点是不便于对数据进行随机访问。
- 二进制文件是以二进制形式存储数据，其优点是便于对数据实行随机访问（相同数据类型的数据所占空间的大小均是相同的，不必在数据之间人为地添加分隔符）。在输入/输出过程中，系统不需要对数据进行任何转换。缺点是数据兼容性差。
- 通常纯文本信息（如字符串）以文本文件形式存储，而将数值信息以二进制文件形式存储。

课堂练习

下列函数中，能够对文件进行写操作的是（　）。

A:get

B:read

C:seekg

D:put

课堂练习

下列函数中，能够对文件进行写操作的是（　）。

A:get

B:read

C:seekg

D:put

答案：D

读写文本文件：
成员函数put()向文件中一次写入一个字节。故本题选D。
成员函数get()从文件中一次读取一个字节。
读写二进制文件：
read()成员函数从文件中读取数据。
write()成员函数向文件中写入数据。
访问文件：
成员函数seekg()，可以设置文件读指针的位置。
成员函数seekp()，可以设置文件的写指针位置。

8.4 随机访问文件

- 如果一个文件只能进行顺序存取操作，则称为顺序文件。典型的顺序文件（设备）是键盘、显示器和保存在磁带上的文件。如果一个文件可以在文件的任意位置进行存取操作，则称为随机文件。磁盘文件就是典型的随机文件。
- 在访问文件的过程中，若严格按照数据保存的次序从头到尾访问文件，则称为顺序访问。
- 在访问文件的过程中，若不必按照数据的存储次序访问文件，而是要根据需要在文件的不同位置进行访问，则称为随机访问。
- 显然，对于顺序文件只能进行顺序访问；对于随机文件既可以进行顺序访问，也可以进行随机访问。

8.4 随机访问文件

类istream中与位置指针相关的函数如下：

(1)移动读指针函数

istream & seekg(long pos);

该函数的功能是将读指针设置为pos，即将读指针移动到文件的pos字节处。

istream & seekg(long offset, ios::seek_dir dir);

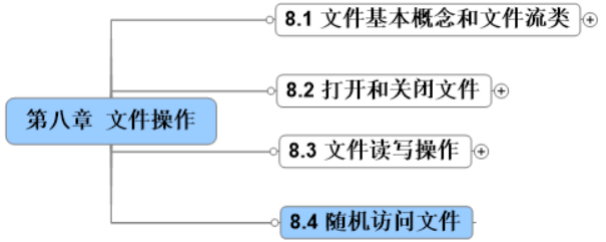
该函数的功能是将读指针按照seek_dir的指示（方向）移动offset个字节，其中seek_dir是在类ios中定义的一个枚举类型。

enum seek_dir {beg=0, cur, end};

seek_dir的常量值含义如下：

- ios::beg：表示流的开始位置。此时，offset应为非负整数。
- ios::cur：表示流的当前位置。此时，offset为正数则表示向后（文件尾）移动，为负数则表示向前（文件头）移动。
- ios::end：表示流的结束位置。此时，offset应为非正整数。

8.4 随机访问文件



(2)返回写指针当前位置的函数

long tellg( );

函数返回值为流中读指针的当前位置。

8.4 随机访问文件

类ostream中与位置指针相关的函数如下：

(1)移动写指针函数

ostream & seekp(long pos);

该函数的功能是将写指针设置为pos，即将写指针移动到文件的pos字节处。

ostream & seekp(long offset, ios::seek_dir dir);

该函数的功能是将写指针按seek_dir指示的方向移动offset个字节。

(2)返回写指针当前位置的函数

long tellp( );

函数的返回值为流中写指针的当前位置。

注意：在类fstream中既提供了操作读指针函数seekg()和tellg()，又提供了操作写指针的函数seekp( )和tellp( )，实际上在文件中这两个指针是同一个指针。

课堂练习

当使用ofstream流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为（　　）。

A:ios::out|ios::binary

B:ios::in|ios::binary

C:ios::out

D:ios::in

课堂练习

当使用ofstream流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为（　　）。

A:ios::out|ios::binary

B:ios::in|ios::binary

C:ios::out

D:ios::in

答案：C

课堂练习

进行文件操作时需要包含头文件（　　）。

A:iostream

B:fstream

C:stdio

D:stdlib

课堂练习

进行文件操作时需要包含头文件（　　）。

A:iostream

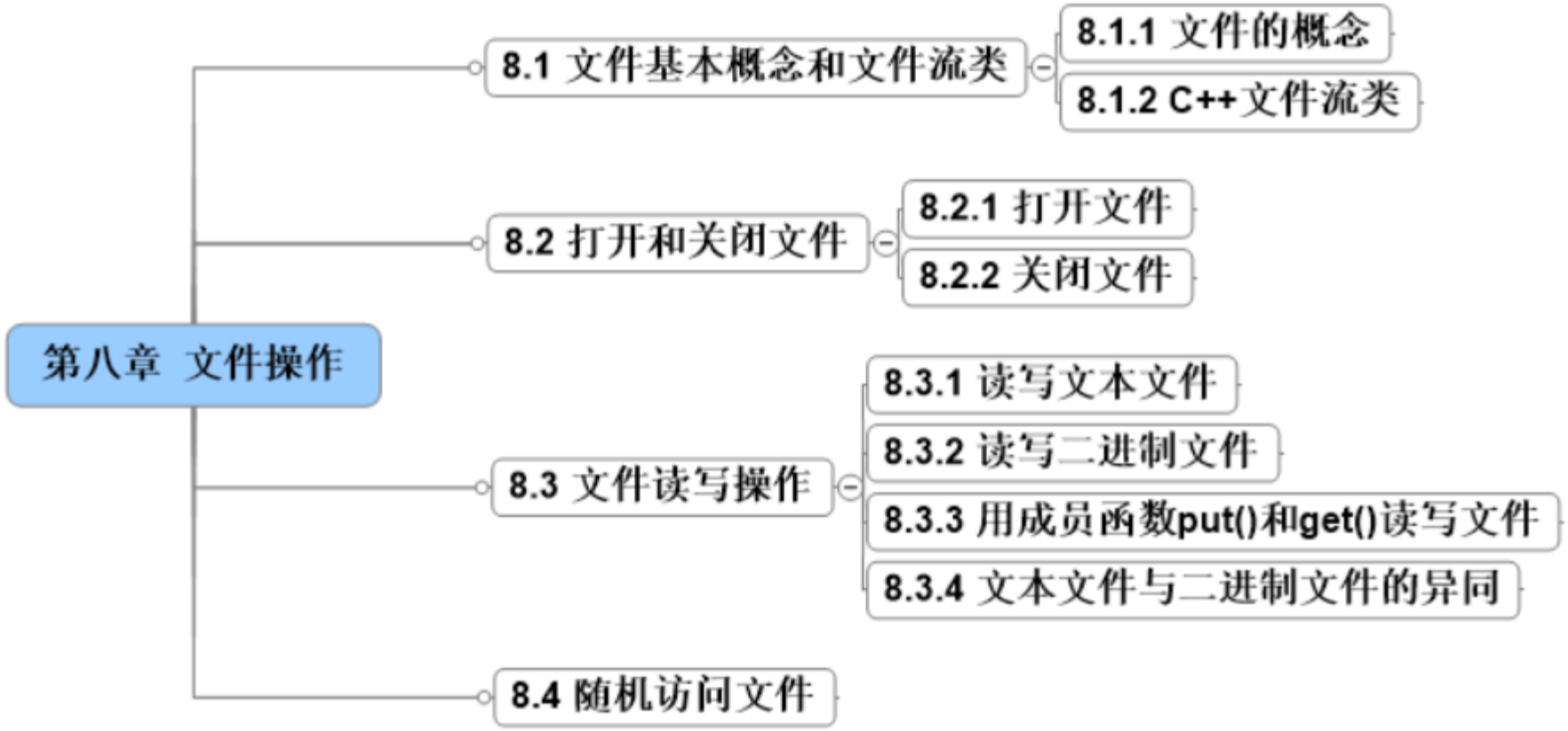
B:fstream

C:stdio

D:stdlib

答案：B

本章总结


第八章 文件操作
8.1 文件基本概念和文件流类
8.1.1 文件的概念
8.1.2 C++文件流类
8.2 打开和关闭文件
8.2.1 打开文件
8.2.2 关闭文件
8.3 文件读写操作
8.3.1 读写文本文件
8.3.2 读写二进制文件
8.3.3 用成员函数put()和get()读写文件
8.3.4 文本文件与二进制文件的异同
8.4 随机访问文件

祝大家顺利通过考试！