Assignment 12 答案

一、概念简答题

1. 分析说明C++语言的流类库中为什么要将ios类作为其派生类的虚基类。

从流类库的基本结构可以看到，ios类是istream 类和ostream 类的基类，从ios头公有派生 istream 和ostream两个类， 而iostream 类通过多重继承istream 类和ostream类而产生的。如果不将ios类作为其派生类的虚基类，可能会产生二义性。

2. 请简要概述文件缓冲区的作用，并结合其回答，程序中为什么要显示的关闭文件?

1. 文件缓冲区是用以暂时存放读写期间的文件数据而在内存区预留的一定空间，使用文件缓冲区 可减少读取硬盘的次数，解决CPU与I/O设备之间读写速度不匹配的问题
2. 如果写入缓冲区后、关闭文件前，程序抛出了异常， fstream 类的析构函数不会被正常调用，则缓冲区内数据不会被写入文件

二、代码编程题

1. 在第九次作业中，我们实现了复数矩阵的运算，本次作业要求增加一些输入输出接口。具体要求如下：

Complex类 需要实现输入输出接口：

```
friend istream &operator >> (istream &input, Complex &C);
friend ostream &operator << (ostream &output, const Complex &C);
```

其中，复数的输入格式为 int+int的形式，第一个int为复数的实部，第二个int为复数的虚部，完成后可以直接进行调用：

```
Complex C;
cin >> C; //控制台输入 "2+3"
cout << C; //控制台输出 "2+3i"
```

Matrix类 需要实现输入输出接口：

```
friend istream &operator >> (istream &input, Matrix<T> &M);
//矩阵输入，第一行为 行数n 列数m，紧接着有n行m列的矩阵元素，例如
//2 2
//1+2 2+3
//3+2 1+1
friend ostream &operator << (ostream &output, const Matrix<T> &M);
//实现矩阵的输出，输出首先为 行数 列数 回车 矩阵内容，例如
//2 2
//1+2i 2+3i
//3+2i 1+1i
```

同时进行文件输入输出测试，定义矩阵文件 matrix_test.txt 其中有如下内容：

```
2 3
1+1 2+1 3+1
2+3 4+2 5+3
```

从matrix_test.txt 中读入矩阵，并输出到新文件中。

注意：利用所学知识进行实现，自行构造其他测试用例。

输入输出接口的代码为：

```
template <class T>
istream& operator >> (istream& input, Matrix<T>& M) {
    int n, m;
    input >> n >> m;
    Matrix<T>t(n, m);
    M = t;
    for(int i = 0; i < M.row; i++){
        for (int j = 0; j < M.col; j++) {
            input >> M[i][j];
        }
    }
    return input;
}
template <class T>
ostream& operator << (ostream& output, const Matrix<T>& M) {
    output << M.row << " " << M.col << endl;
    for (int i = 0; i < M.row; i++) {
        for (int j = 0; j < M.col; j++) {
            output << M[i][j];
            output << " ";
        }
        output << endl;
    }
    return output;
}
```

文件输入输出调用如下：

```
ifstream in_file("matrix_test.txt", ios::in);
in_file >> m1;
in_file.close();
ofstream out_file("matrix.txt", ios::out);
out_file << m1;
out_file.close();
```

注意：同时需要增加Complex类的输出输出接口

```
istream& operator >> (istream& input, Complex& C) {
    string s;
    input >> s;
    int pos = s.find("+", 0);
    string sTmp = s.substr(0, pos);
```

```
    C.real = atoi(sTmp.c_str());
    sTmp = s.substr(pos + 1, s.length() - pos - 1);
    C.image = atoi(sTmp.c_str());
    return input;
}
ostream& operator << (ostream& output, const Complex& C) {
    output << C.real << "+" << C.image << "i";
    return output;
}
```

2. 给定一个文件如下

```
12345678910111213...
```

为1-100数字的拼接，首先你需要生成这样的一个文件，然后读取文件，将以0结尾的数字输出出来。例如，我们可以通过如下方式输出数字10：

```
char t[20] = "\0";
file.seekg(9, ios::beg);
file.read(t,2);
cout << t;
```

注意：利用文件的随机存取来实现。

答案：首先是文件的生成

```
ofstream out_file("number.txt", ios::out);
for (int i = 1; i <= 100; i++)
    out_file << i;
out_file.close();
```

然后是随机读取部分：

```
std::ifstream in("number.txt");
char t[20] = "\0";
in.seekg(9, std::ios::beg);
in.read(t, 2);
std::cout << t << ' ';
for (int i = 0; i < 8; ++i) {
    in.seekg(18, std::ios::cur);
    in.read(t, 2);
    std::cout << t << ' ';
}
in.seekg(18, std::ios::cur);
in.read(t, 3);
std::cout << t << std::endl;
```

3. 某课程G需要将考试成绩录进系统并进行相应的保存，需要实现一个简单的成绩表管理系统

一条成绩的记录包括学号，姓名，性别和分数，例如“1 小明 男 88”。

1. 我们需要从 **键盘** 录入成绩的记录信息，然后保存到文件 a 当中。
2. 从文件 a 中读出成绩信息。
3. 输出成绩前3名的学生信息，输出到新文件 b 中。
4. 对于男女生各计算平均分，再将成绩在平均分以下的记录输出到一个新文件 c 中。
5. 增加几条新的补考记录，对分数按 90% 折算录进系统。
6. 输出数据到文件 a 中，完成表格的更新。

要求：设计成绩表为一个类，通过重载 << 和 >> 来实现成绩表的输入/输出。

代码如下：

```
enum Sex {
    MALE, FEMALE
};

class Grade {
private:
    std::string id;
    std::string name;
    Sex sex;
    double score;
public:
    Grade() {
        sex = MALE;
        score = 0;
    };

    Grade(const std::string& _id, const std::string& _name, Sex _sex, double
_score){
        id = _id;
        name = _name;
        sex = _sex;
        score = _score;
    }

    bool operator < (const Grade& g) const{
        return score > g.score;
    }

    Sex getSex() const{
        return sex;
    }

    double getScore() const{
        return score;
    }

    void fix(){
        this->score = this->score * 0.9;
    }
```

```
    friend std::istream& operator >> (std::istream& in, Grade& g) {
        in >> g.id >> g.name;
        std::string inputsex;
        in >> inputsex;
        if (inputsex == "MALE") {
            g.sex = MALE;
        } else if (inputsex == "FEMALE") {
            g.sex = FEMALE;
        }
        return in >> g.score;
    }

    friend std::ostream& operator << (std::ostream& out, const Grade& g){
        out << g.id << ' ' << g.name << ' ';
        switch (g.sex) {
            case MALE:
                out << "MALE"; break;
            case FEMALE:
                out << "FEMALE"; break;
        }
        return out << ' ' << g.score;
    }
};


class GradeTable {
private:
    std::vector<Grade> gradeVector;
public:
    GradeTable(){
        gradeVector.clear();
    };
    void mysort() {
        sort(gradeVector.begin(),gradeVector.end());
    }

    Grade& operator[] (int n) {
        return this->gradeVector[n];
    }

    double getSexAvg(Sex sex) const {
        int n = 0;
        double score = 0;
        for (const Grade& g: gradeVector) {
            if (g.getSex() == sex) {
                ++n;
                score += g.getScore();
            }
        }
        return score / n;
    }

    int getSize() const {
        return this->gradeVector.size();
    }

    void addGrade(const Grade& g) {
        this->gradeVector.push_back(g);
```

```
    }

    friend std::istream& operator >> (std::istream& in, GradeTable& gt) {
        gt.gradeVector.clear();
        int num = 0, i = 0;
        in >> num;
        while (i < num) {
            Grade g;
            in >> g;
            gt.gradeVector.push_back(g);
            i++;
        }
        return in;
    }

    friend std::ostream& operator << (std::ostream& out, const GradeTable&gt){
        out << gt.getSize() << std::endl;
        for (const Grade& g: gt.gradeVector) {
            out << g << std::endl;
        }
        return out;
    }
};
```

测试用例：

原a.txt

```
7
1 xiaoming MALE 61
2 xiaohong FEMALE 61
3 xiaoqiang MALE 100
4 xiaozhang MALE 56
5 xiaotao MALE 87
6 xiaoqi MALE 90
7 xiaoyi FEMALE 98
```

b.txt

```
3
3 xiaoqiang MALE 100
7 xiaoyi FEMALE 98
6 xiaoqi MALE 90
```

c.txt

```
4
3 xiaoqiang MALE 100
7 xiaoyi FEMALE 98
6 xiaoqi MALE 90
5 xiaotao MALE 87
```

更新后的a.txt

```
8
3 xiaoqiang MALE 100
7 xiaoyi FEMALE 98
6 xiaoqi MALE 90
5 xiaotao MALE 87
1 xiaoming MALE 61
2 xiaohong FEMALE 61
4 xiaozhang MALE 56
8 xiaohu MALE 90
```