Introduction to C++ (Season 2)

Unit 7: File Input and Output

第7单元:出入虽同趣，所向各有宜— 文件输入输出

Section 07 : More on Binary IO

第07节： 关于二进制读写的更多内容

Binary Array I/O (数组的二进制读写)

❖ Example: →

- write an array of double values to a binary file
- read it back from the file.

```
double array[SIZE] = {3.4, 1.3, 2.5, 5.66, 6.9};

//binaryio.write(reinterpret_cast<char*>(&array), sizeof(array));

binaryio.write(reinterpret_cast<char*>(array), sizeof(array));
```

→ BinaryArrayIO

Binary Object I/O (对象二进制读写)

- writes student records into a text file. (将学生记录写入文本文件)
  - first name middle name initial last name score.
- How to write a object to file?(如何将记录写入文件)
  - write the fields to the file separately. (仅写数据域，而函数则无关)
- Example: write/read an object of the Student class.

Student
-firstName: string
-mi: char
-lastName: string
-score: double
+Student()
+Student(firstName: string, mi: char, lastName: string, score: int)
+getFirstName(): string
+getMi(): char
+getLastName(): string
+getScore(): int
+setFirstName(s: string): void
+setMi(ch: char): void
+setLastName(s: string): void
+setScore(score: int): void

The first name of this student.
The middle name initial of this student.
The last name of this student.
The score of this student.

Constructs a default Student object.
Constructs a student with specified first name, mi, last name, and score
Returns the first name of this student.
Returns the mi of this student.
Returns the last name of this student.
Returns the score of this student.
Sets a new first name of this student.
Sets a new mi of this student.
Sets a new last name of this student.
Sets a new score for this student.

Binary Object I/O, cont.

```
void displayStudent(Student student) {
  cout << student.getFirstName() << " ";
  cout << student.getMi() << " ";
  cout << student.getLastName() << " ";
  cout << student.getScore() << endl;
}

int main() {
  fstream binaryio; // Create stream object
  binaryio.open("object.dat", ios::out |
                              ios::binary);

  Student student1("John", 'T', "Smith", 90);
  Student student2("Eric", 'K', "Jones", 85);

  binaryio.write(reinterpret_cast<char *>
    (&student1), sizeof(Student));
  binaryio.write(reinterpret_cast<char *>
    (&student2), sizeof(Student));
```

```
  // Read student back from the file
  binaryio.open("object.dat", ios::in |
                              ios::binary);

  Student studentNew;

  binaryio.read(reinterpret_cast<char *>
    (& studentNew), sizeof(Student));

  displayStudent(studentNew);

  binaryio.read(reinterpret_cast<char *>
    (& studentNew), sizeof(Student));

  displayStudent(studentNew);

  binaryio.close();

  return 0;
}
```