**Course Syllabus**

**课程编号：071232 学分：3.5**

**C++面向对象程序设计**

**(Object-oriented Programming Using C++)**

**课程性质：**subject/required

**适用专业：**Computer science and technology , network engineering

**学时分配：**Total periods: 56, including theoretical periods: 32, and experimental periods:24.

**先行、后续课程情况：**

Preparing courses: Programming Design Using C;

Follow-up courses: Operational System, Software Engineering

**教材：**Richard Johnsonbaugh, Martin Kalin. *Object-Oriented Programming using C++* (Second Edition).北京：清华大学出版社，2005.8

**参考书目：**

1．Goran Svenk．Object-Oriented Programming using C++ for Engineering and Technology（影印版）．清华大学出版社,2003.10

2．Bjarne Stroustrup．The C++ Programming Language（Special Edition）．北京：机械工业出版社,2002.7

3．Bruce Eckel．Thinking in C++（Second Edition）．北京：机械工业出版社,2006.10

4．郑莉．C++语言程序设计（第3版）．北京：清华大学出版社，2005.7

5．钱能．C++程序设计教程（第2版）．北京：清华大学出版社，2005.8

6．谭浩强．C++面向对象程序设计．北京：清华大学出版社，2006.1

7．Walter Savitch著，佟俐鹃等改编．C++面向对象程序设计（第6版）．北京：清华大学出版社，2008.3

**一、课程的目的与任务***C++ Object-oriented Programming* is a compulsory course for undergraduates of computer science and technology. Object-oriented Programming is a software-developing method, which absorbed the beneficial concepts and effective ways in software engineering field. Being abstract, successive and multi-stated, the object-oriented programming can help people develop modularizing and data-highly- abstracted programs with other characters such as concealment, copying, easily corrected and easily extended.

This course mainly introduces the object-oriented programming and basic concepts of C++, with the object-oriented mechanism in C++ as the main part. Through lots of programming cases and concerned exercises, learners will grasp the functions of object-oriented programming gradually, and other involved basic knowledge and skills.

**二、课程的基本要求**

1. Having a general understanding of the basic concepts of object-oriented programming

and primary thoughts of the usage of object-oriented programming.

1. Being able to use C++ to practice object-oriented programming design in a general way.
2. Having a primary grasp of the usage of Visual C++ in object-oriented programming.
3. Adopting Chinese and English as the teaching languages and using English version

teaching materials.

1. Teaching language must be English. Both English and Chinese can be used when

explain the concepts, and the special terms must be explained in English.

1. With comparatively more concepts and special terms, special attention should be focused

on the explanation of both basic concepts and basic theories of computer communication and network. Pay attention to the accuracy of English words.

1. Computer science and networks enjoy a fast development with new technology appeared

continually, thus we should pay attention to the introduction of new technology.

1. Emphasizing practice. Special attention should be given to the process of applying theory

to practice and the experimental periods.

1. Please notice that English is just a teaching and learning tool. Our aim is to use English to

learn majors not to learn English itself. Teaching English is not our aim.

1. Only English or only Chinese is not the teaching way in class. At the very beginning, the

English proportion should not be too high, but we can raise the proportion gradually. So

does the communication between teacher and students. Generally speaking, English

proportion should not be lower than 50 percent.

1. Owing to the adoption of bilingual teaching, learning periods is comparatively

inadequate, therefore we should pay attention to the efficiency of teaching and reminding students of preview and review to keep up with the teaching process.

1. Suggesting teachers provide students with the Chinese meanings of the English special

terms thus to avoid too much time and energy are wasted in consulting dictionary.

1. English should be used on blackboard and in homework as well as examination.

**三、课程教学内容**

**第一章：OBJECT-ORIENTED PROGRAMMING**

**1.该章的基本要求与基本知识点**

(1) Understanding Object-Oriented Programming, Object-Oriented and Procedural Programming Relationships;

(2) Knowing the concept of Classes and Abstract Data Types, the Client/Server Model and message passing;

(3) Having a primary understanding of inheritance, polymorphism, interfaces and

components.

**2.要求学生掌握的基本概念、理论、原理**

Object-Oriented Programming, Object-Oriented and Procedural Programming Relationships,

Classes and Abstract Data Types, the Client/Server Model and message passing, understanding of inheritance, polymorphism, interfaces and components.

**3.教学重点与难点**

**Teaching emphases:** Object-Oriented Programming; Inheritance and polymorphism.

**Teaching difficulties:** Classes and abstract data Types; Inheritance and polymorphism.

**4.实验：**

**Experimental items:** Grasping Visual C++ programming situation. (2 periods)

**Experimental contents:** Designing simple I/O programs.

**第二章：FROM C TO C ++**

**1.该章的基本要求与基本知识点**

(1) Understanding namespaces, C++ Input/Output and Files;

(2) Knowing C++ Features, the Type string, functions and the new and delete operators;

(3) Having a primary understanding of C++ Input/Output.

**2.要求学生掌握的基本概念、理论、原理**

namespaces, C++ Input/Output and Files, C++ Features, the Type string, functions and the new and delete operators.

**3.教学重点与难点**

**Teaching emphases:** Namespaces; C++ Input/Output; C++ Features.

**Teaching difficulties:** Namespaces; C++ Input/Output.

**Experimental items:** Using function to design modularization program. (2 periods)

**Experimental contents:** Using if sentence , while sentence ,do while sentence and for sentence to program; Designing overloaded functions.

**第三章：CLASSES**

**1.该章的基本要求与基本知识点**

(1) Understanding and mastering classes and objects;

(2) Mastering the use of constructors and the destructor;

(3) Understanding the feature of data members, methods and pointers to objects.

**2.要求学生掌握的基本概念、理论、原理**

classes and objects, constructors and the destructor, data members, methods and pointers to objects.

**3.教学重点与难点**

**Teaching emphases:** Classes and objects; Constructors and the destructor; Data members and methods.

**Teaching difficulties:** Constructors and the destructor; Pointers to objects.

**4.实验：**

**Experimental items:** Using object-oriented programming to design programs.(4 periods)

**Experimental contents:** Definition and application of class, object, constructor function and destructor function

**第四章：INHERITANCE**

**1.该章的基本要求与基本知识点**

(1) Mastering the basic concepts and syntax of inheritance;

(2) Understanding the protected members, constructors and destructors under inheritance;

(3) Having a general grasp of multiple inheritance.

**2.要求学生掌握的基本概念、理论、原理**

the basic concepts and syntax of inheritance, the protected members, constructors and destructors under inheritance, a general grasp of multiple inheritance.

**3.教学重点与难点**

**Teaching emphases:** Protected members; Constructors and the destructors under

inheritance; Multiple Inheritance.

**Teaching difficulties:** Constructors and the destructors under Inheritance; Multiple

inheritance.

**4.实验：**

**Experimental items:** Deriving a new class and reusing code by inheritance. (4 periods)

**Experimental contents:** Using public, private, protected; Inheritance mode and order.

**第五章：POLYMORPHISM**

**1.该章的基本要求与基本知识点**

(1) Knowing the Run-Time versus Compile-Time binding in C++;

(2) Understanding and mastering the name overloading, name overriding, and name hiding;

(3) Knowing the abstract base classes and run-time type identification.

**2.要求学生掌握的基本概念、理论、原理**

the Run-Time versus Compile-Time binding in C++, name overloading, name overriding, and name hiding, abstract base classes and run-time type identification.

**3.教学重点与难点**

**Teaching emphases:** Name overloading, name overriding, name hiding; Abstract base classes; Definition of static member; Definition of friend Functions.

**Teaching difficulties:** Definition and application of copy constructor functions; Definition of static member; Definition of friend Functions.

**Experimental items:** Learning how to use virtual functions and polymorphism to design programs by Object-Oriented Programming. (4 periods)

**Experimental contents:** Stating virtual function by virtual and understanding the effect of virtual functions

**第六章：OPERATOR OVERLOADING**

**1.该章的基本要求与基本知识点**

(1) Understanding the basic operator overloading and operator overloading using Top-Level Functions;

(2) Understanding and mastering friend Functions;

(3) Understanding the overloading of the input and output operators, the assignment operators, some special operators and memory management operators.

**2.要求学生掌握的基本概念、理论、原理**

the basic operator overloading and operator overloading using Top-Level Functions, friend Functions, the overloading of the input and output operators, the assignment operators, some special operators and memory management operators.

**3.教学重点与难点**

**Teaching emphases:** Basic operator overloading; Friend functions; Overloading the input and output operators; Overloading the assignment operators; Overloading some special operators.

**Teaching difficulties:** Friend functions; Overloading the input and output operators.

**4.实验：**

**Experimental items:** Writing operator overloading functions. (2 periods)

**Experimental contents:** Programming to realize the two complex numbers addition and subtraction using operator overloading.

**第七章：TEMPLATES AND THE STANDARD TEMPLATE LIBRARY**

**1.该章的基本要求与基本知识点**

(1) Understanding the concept of template, template classes in a parameter list;

(2) Understanding the function-style parameters, standard template library (STL).

**2.要求学生掌握的基本概念、理论、原理**

concept of template, template classes, function-style parameters, standard template library.

**3.教学重点与难点**

**Teaching emphases:** Template; Template classes in a parameter list; Function-Style parameters.

**Teaching difficulties:** Template classes in a parameter list; Function-Style parameters; Standard Template Library (STL).

**Experimental items:** Template functions and template classes. (2 periods)

**Experimental contents:** Designing the stack class template.

**第八章：THE C++ INPUT/OUTPUT CLASS HIERARCHY**

**1.该章的基本要求与基本知识点**

(1) Knowing the classes ios\_base and basic\_ios, the high-level input/output classes (basic\_istream; basic\_ostream; basic\_iostream);

(2) Knowing the file input/output classes (basic\_ofstream; basic\_ifstream; basic\_fstream), the character stream input/output classes (basic\_ostringstream; basic\_istringstream; basic\_stringstream);

(3) Knowing the buffer classes (basic\_streambuf; basic\_filebuf; basic\_stringbuf).

**2.要求学生掌握的基本概念、理论、原理**

ios\_base and basic\_ios, the file input/output classes, the buffer classes.

**3.教学重点与难点**

**Teaching emphases:** The high-level input/output classes (basic\_istream; basic\_ostream; basic\_iostream); Manipulators; The file input/output classes (basic\_ofstream; basic\_ifstream; basic\_fstream).

**Teaching difficulties:**The high-level input/output classes (basic\_istream; basic\_ostream; basic\_iostream).

**4.实验：**

**Experimental items:** Understanding the I/O stream and format manipulators. Mastering the usage of file stream. (2 periods)

**Experimental contents:** Designing a program including the standard I/O and the file I/O.

**5.习题课安排：**

The summary of class ,inheritance, polymorphism , operator overloading and the I/O class; exercises of class definition , inheritance, name overloading, name overriding, abstract base classes ,friend function, operator overloading , template classes, the I/O classes and etc.

**第九章：OBJECT-ORIENTED PROGRAMMING IN THE MICROSOFT FOUNDATION CLASSES**

**1.该章的基本要求与基本知识点**

(1) Knowing the Windows Programming in MFC;

(2) Understanding the document/view architecture in MFC;

(3) Knowing the Component Object Model.

**2.要求学生掌握的基本概念、理论、原理**

Windows Programming, the document/view architecture in MFC, Component Object Model.

**3.教学重点与难点：**

**Teaching emphases:** The document/view architecture in MFC.

**Teaching difficulties:** The document/view architecture in MFC; The Component Object Model(COM).

**4.实验：**

**Experimental items:** Learning how to use MFC to design programs by Object-Oriented Programming. (2 periods)

**Experimental contents:** Designing a small Student MIS depend on the Oriented-Object Programming by Visual C++.

**四、课程学时数安排**

**课程学时数安排**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **内 容** | **课堂教授** | **习题课** | **上 机** | **小 计** | **课外学时** |
| 第一章 | 2 | 0 | 2 | 2 | 2 |
| 第二章 | 4 | 0 | 2 | 6 | 8 |
| 第三章 | 6 | 0 | 4 | 10 | 10 |
| 第四章 | 4 | 0 | 4 | 8 | 10 |
| 第五章 | 4 | 0 | 4 | 8 | 6 |
| 第六章 | 4 | 0 | 2 | 8 | 8 |
| 第七章 | 4 | 0 | 2 | 6 | 6 |
| 第八章 | 1 | 2 | 2 | 5 | 4 |
| 第九章 | 1 | 0 | 2 | 1 | 2 |
| 合计 | 30 | 2 | 24 | 56 | 56 |

**五、教学大纲编制说明**

本大纲是遵照教务处2009版计算机科学与技术专业教学计划中的要求，以《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)高等教育出版社，2006年9月》为指导思想，借鉴兄弟院校同类课程教学大纲，集任课教师多年的教学经验编制而成。本课程安排了24学时的实验，2周的课程设计，旨在加强学生的编程能力。

**制订人： 李兰 审核人：**