课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	C/C++ 程序设计 C/C++ programming Design
2.	**授课院系 Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号 Course Code**	CS205
4.	**课程学分 Credit Value**	3
5.	**课程类别 Course Type**	Major elective course
6.	**授课学期 Semester**	第二学年秋季和春季学期 Fall and spring semester , the second year
7.	**授课语言 Teaching Language**	English
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师） Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）**	Feng Zheng, Assistant Professor, Department of Computer Science and Engineering, zhengf@sustech.edu.cn Shiqi Yu, Associate Professor, Department of Computer Science and Engineering, yusq@sustech.edu.cn
9.	**实验员/助教、所属学系、联系方式（请列出本课所有教辅人员） Tutor/TA(s), Contact (Please list all)**	Qimei Liao, Department of Computer Science and Engineering, liaoqm@mail.sustech.edu.cn
10.	**选课人数限额(可不填) Maximum Enrolment （Optional）**	

11.	授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
	学时数 Credit Hours	32		32		64

12.	先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	计算机导论 Introduction to Computer
13.	后续课程、其它学习规划 Courses for which this course is a pre-requisite	数据结构 Data Structure 算法分析与设计 Algorithm Analysis and Design
14.	其它要求修读本课程的学系 Cross-listing Dept.	

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

C/C++是计算机科学与技术专业一门专业基础课程。其中 C 是一种面向问题的通用程序设计语言，具有语言简洁、类型丰富、结构完整、表达力强、直接操作内存单元、适用于模块化结构等特点。C 语言既具有高级语言的优点，又具有低级语言的许多特点。其中 C++是面对对象开发方法，从 C 语言扩展而来。吸收了软件工程领域有益的概念和有效方法，它把数据和对数据的操作封装起来，集抽象性、封装性、继承性与多态性于一体，可以帮助人们开发出模块化、数据抽象程度高的、信息隐蔽好的、可复用、易修改、易扩充等特性的程序。
该课程包括两个部分，第一部分为 C++中一般性设计原理，是 C 的部分；第二部分为面向对象部分。通过该课程的学习，学生可以为数据结构，算法设计与分析等课程打下坚实的基础。

C/C++ is a professional foundation course in computer science and technology. C is a general-purpose programming language oriented to problems. It has the characteristics of simple language, rich types, complete structure, strong expressiveness, direct operation of memory units, and application for modular structure. The C language has both the advantages of a high-level language and many features of a low-level language. While, C++ is an object-oriented development method that extends from the C language. It absorbs the useful concepts and effective methods in the field of software engineering. It encapsulates data and operations on data, and integrates abstraction, encapsulation, inheritance and polymorphism to help people develop programs with high modularity, high level of data abstraction, information concealed, reusable, easy to modify, and easy to expand.
The course consists of two parts, the first part is the general design principle in C++, which is part of C; the second part is the object-oriented part. Through this course, students can build a solid foundation for the courses of data structure, and algorithm design and analysis.

16. 预达学习成果 Learning Outcomes

完成该课程，学生能够做到：
能够解释 C/C++程序语言的工作原理
能够阅读 C/C++程序，能够找出其中错误并修正错误
能够分析实际问题，构建 C/C++程序解决该问题
具有专业化的编程态度与习惯
面对实际问题时，具有编程思维

Upon completion of this course, students should be able to:
Explain how an existing C or C++ program works
Discover errors in a C or C++ program and describe how to fix them
Analyze a problem and construct a C or C++ program that solves it
Professional programming attitude and habits

	Programming thinking
17.	**课程内容及教学日历 （语言与授课语言一致，例：如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）** **Course Contents (in Parts/Chapters/Sections/Weeks. In consistency with instructional language. Please notify name of instructor for course section(s), if this is a team teaching or module course.)**

第一周：导论

- 课程介绍
- C/C++背景介绍
- C/C++基本概念

[实验课]

- 作业：介绍实验开发平台、编译工具并且指导学生成功安装各种工具。
- 作业：完成一个大整数乘法计算
- 考察目标：考察学生的 C++入门知识，学会编译和链接等。

第二周：处理数据

- 简单变量和 const
- 浮点数
- C++算术运算符

[实验课]

- 作业内容：参考 Linux 下的计算器 bc，实现一个任意精度的计算器
- 考察目标：考察学生对 C/C++的基本语法了解程度，学生应当在本次作业中了解到 Java 与 C/C++之间的区别，以及深刻理解数据类型的精度问题。

第三周：循环、条件判断

- for 循环
- while 循环
- 循环的应用
- if 语句
- 逻辑运算符与特殊运算符？
- switch 语句

[实验课] 如上。

第四周：复合类型

- 数组
- 字符串和 string 类
- 结构体
- 指针类型
- 动态内存分配基础

[实验课]

- 作业内容：定义一个 struct 用于描述向量，并实现两个长向量的乘法。

- 考察目标：考察学生是否掌握结构体和动态内存分配。

第五周：函数

- 函数与数组
- 函数与结构
- 函数与字符串
- 函数与指针

[实验课] 如上。

第六周：函数探幽

- C++内联函数
- 引用变量
- 函数重载
- 函数模板

[实验课]

- 作业内容：将上一个作业扩展到矩阵乘法计算。
- 考察目标：考察函数的重栽、引用变量等知识点。

第七周：程序编译与连接

- 单独编译
- 存储持续性、作用域和链接性
- 名称空间
- ARM Linux 上编程
- ARM Linux 编程与 PC 平台比较

[实验课]

- 作业内容：在 ARM Linux 上实现一个矩阵运算，跟 PC 上的运行速度进行对比。
- 考察目标：考察学生不同编程平台的了解。

第八周：程序的运行效率

- 类 cv::Mat 中的 ROI 设置方法
- 内存读写与速度
- SIMD 指令
- 多 CPU 并行
- GPU

[实验课]

- 作业内容：实现矩阵乘法，并与 cblas 库对比运行效率。
- 考察目标：考察学生计算机体系结构的底层理解。

第九周：对象和类

- 过程性编程与面向对象编程比较

- 抽象和类
- 类的构造和析构函数
- this 指针
- 类作用域

[实验课]

- 作业内容：给定包含 UTF-8 字符的文件，寻找文件中最高频率的字符所属的 UTF-8 字符子集。
- 考察目标：考察学生对字符编码机制和 C/C++文件 IO 了解程度。

第十周：使用类

- 运算重载符与示例
- 友元
- 重载运算符：作为成员函数还是非成员函数
- 类的自动转换和强制类型转换

[实验课] 如上。

第十一周：类和动态内存分配

- 动态内存和类
- 改进后的 string 类
- 在构造函数中使用 new 时应该注意事项
- 有关返回对象声明
- 使用指向对象的指针

[实验课]

- 作业内容：实现一个支持 UTF-8 字符的字符串类。
- 考察目标：考察学生对 C++面向对象编程的掌握程度。

第十二周：类继承

- 一个简单的基类
- 继承：is-a 关系
- 多态共有继承
- 静态联编和动态联编
- 访问控制：protected
- 继承和动态内存分配

[实验课] 如上。

第十三周：C++中的代码重用

- 包括对象成员的类
- 私有继承
- 多重继承
- 类模板

[实验课]

- 作业内容：对上一个作业中中支持 UTF-8 字符的字符串类进行扩展，使其支持 C++原生的运算符。
- 考察目标：考察学生对 C++运算符重载机制的掌握程度。

第十四周：友元和异常

- 友元
- 嵌套类
- 异常

[实验课] 如上。

第十五周：string 类和标准模板库

- string 类
- 智能指针模板类
- 标准模板库
- 泛型编程
- 函数对象

[实验课]

- 作业内容：实现一个 3 维立方体类，该类拥有多个类成员函数并支持多个运算符。
- 考察目标：考察学生对 C++类和对象，运算符重载机制的掌握程度。

第十六周：总结和复习

[实验课] 复习、答疑。

Weak 1: Introduction

- Class introduction
- C/C++ background
- C/C++ basic concepts

[Lab 1]: Introduce IDE and compiler tools and help student successfully install these platforms and tools

Weak 2: Dealing with Data

- Simple variables and constants
- Floating point number
- C ++ arithmetic operator

[Lab 2]:

- Content：Please implement a much better calculator than that in the previous assignment. More features which can be found from the calculator BC in Unix-like systems.
- Purpose：Students will learn the basic concepts of C/C++. And students can learn the differences between Java and C/C++.

Week 3: Loops, Branch Statements and Expressions

- for loops
- while loops
- loop applications
- if statement
- Logical operators and special operator: ?
- switch statement

[Lab 3]:

- The same with above.

Week 4: Compound Types

- Array
- String and string class
- Structure
- Pointer type
- Dynamic memory management (basic)

[Lab 4]:

- Content: Design a struct for 1D vectors and implement a function for dot product of two vectors.
- Purpose: Students should master the knowledge of struct and pointer.

Week 5: Function

- Functions and arrays
- Function and structures
- Function and strings
- Function and pointers

[Lab 5]: The same with above.

Week 6: Adventures in Function

- C++ inline function
- Reference variable
- Function overloading
- Function template

[Lab 6]:

- Content: To improve the previous assignment, implement a struct for a matrix, implement a function for matrix multiplication.

- Purpose: Students should master the knowledge of function overloading and reference variable.

Week 7: Memory Modules and Namespace

- Compiled separately
- Storage duration, scope and linkage
- Namespace
- Programming on ARM boards
- Comparisons between ARM and PC.

[Lab 7]:

- Content: To compare the matrix multiplication on PC and ARM. To improve the performance on ARM.
- Purpose: Students should master the knowledge of different platforms.

Week 8: Program Efficiency

- ROI in class cv::Mat
- Memory access and speed
- SIMD
- Working with multiple CPU cores
- GPU

[Lab 8]:

- Content: To improve the matrix multiplication and compare it with cblas or some other similar math libraries.
- Purpose: Students should understand the structure of computer.

Week 9: Objects and Classes

- Comparison between procedural programming and object-oriented programming
- Abstraction and class
- Class constructors and destructors
- The this pointer
- Class scope

[Lab 9]:

- Content: Given a file which includes UTF-8 characters, implement a program which can find the subset of the highest frequent character in a file with UTF-8 characters.
- Purpose: Students should master the knowledge of string, encoding and file I/O in C++.

Week 10: Working with Classes

- Operator overloading and examples
- Introducing friend

- Overloaded operator: as a member function or a non-member function
- Automatic conversion and type cast for class

[Lab 10]: The same with above

Week 11: Class and Dynamic Memory Allocation

- Dynamic memory and class
- The improved string class
- Things to remember when using new
- Observations about returning objects
- Using pointers to objects

[Lab 11]:

- Content: Implement a string class which supports UTF-8 characters.
- Purpose: Students will learn the object-oriented programming in C++.

Week 12: Class Inheritance

- A simple base class
- Inheritance: an is-a relationship
- Polymorphic public inheritance
- Static and dynamic binding
- Access control: protection
- Inheritance and dynamic memory allocation

[Lab 12]: The same with above.

Week 13: Reusing Code in C++

- Classes with object members
- Private inheritance
- Multiple inheritance
- Class templates

[Lab 13]:

- Content: Extends the string class in assignment so it can support the native operators in C++
- Purpose: Students will should master the operator overloading in C++.

Week 14: Friends and Exceptions

- Friend
- Nested class
- Exceptions

[Lab 7]: The same with above.

Week 15: String Classes and Standard Template Library

- String class
- Smart pointer template class
- Standard template library
- Generic programming
- Function objects

[Lab 15]:

- Content: Implement a 3D box class which includes several member functions and supports native operators in C++.
- Purpose: Students should master object-oriented programming and the operator overloading in C++.

Week 16: Summary and Review

[Lab 16]:

- The same with above
- Review and answer questions

18. 教材及其它参考资料 Textbook and Supplementary Readings

C++ Primer Plus

Sixth Edition

ISBN-13: 978-0-321-77640-2

Stephen Prata

教学评估 ASSESSMENT

19.	评估形式 Type of Assessment	评估所需时间 Duration	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
	出勤 Attendance		/		
	课堂表现 Class Performance		5%		
	小测验 Quiz		5%		
	课程项目 Projects		60%		
	平时作业 Assignments		30%		

	期中考试 **Mid-Term Test**		/		
	期末考试 **Final Exam**		/		
	期末报告 **Final Presentation**		/		
	其它（可根据需要改写以上评估方式） **Others (The above may be modified as necessary)**		/		

课程审批 REVIEW AND APPROVAL	
20.	本课程设置已经过以下责任人/委员会审议通过 **This Course has been approved by the following person or committee of authority**