# 操作系统：

先修课程：程序设计与C语言、数据结构、计算机组成原理

课程简介：操作系统课程主要讲述操作系统对计算机系统资源进行管理的基本原理和实现方法。具体内容包括进程管理、存储器管理、文件系统、I/O设备管理、死锁等。课程通过实例分析，帮助学生掌握操作系统的基本理论和实现方法，了解现代操作系统的新概念、新技术和新发展。课程实验环节旨在培养学生分析、设计和开发操作系统的基础能力。

#### 建议：

1. 确保基础扎实

* C语言与算法：操作系统课程会涉及到大量的编程，特别是底层代码。课程实验主要以xv6操作系统为基础，xv6用C语言编写，所以你应当对C语言及一些相应算法熟练运用。
* 计算机组成原理：理解计算机的基本结构和工作原理是学习操作系统的基础。但这两门课关联性较小，因而我校在同一学期开课。
* 数据结构：操作系统中的很多算法和机制都依赖于数据结构。复习数据结构的课程内容，确保你能熟练运用各种数据结构解决问题。

2. 时间管理

* 操作系统课程通常包含大量的实验和编程作业，需要投入相当的时间和精力。确保你有足够的时间来完成这些任务，避免因为时间不足而影响学习效果。

3. 资源准备

* 教材和参考书：购买或借阅推荐的教材，如《现代操作系统》、《操作系统: 精髓与设计原理》和《操作系统概念》。这些书籍是学习操作系统的经典资源。我校使用的教材是《现代操作系统》。
* 实验环境：熟悉并安装必要的实验工具和软件，最重要的是虚拟机（VirtualBox）和VSCode。虚拟用于建立Linux环境，而VSCode用于对虚拟机操作系统资源的远程访问和修改。

4. 心理准备

* 操作系统课程可能会比较难，但它对计算机科学和技术专业学生来说至关重要。保持积极的心态，勇于面对学习中的困难和挑战。

# 计算机网络

#### 先修课程：

计算机系统结构、计算机程序设计、操作系统

课程简介：  
计算机网络课程是计算机科学与技术专业的一门必修课，主要内容包括：

1. 因特网与TCP/IP模型：介绍因特网的基本组成、网络边缘、核心、分组交换技术、协议分层思想等。
2. 应用层：涵盖网络应用的基本原理、Web和HTTP、FTP、电子邮件（SMTP、POP、IMAP）、DNS服务、P2P应用、Socket编程等。
3. 传输层：讲解传输层服务、多路复用与解多路复用、UDP、TCP协议、可靠数据传输、拥塞控制等。
4. 网络层：包括虚电路与数据报网络、路由器、IPv4、DHCP、NAT、IPv6、路由算法（DV与LS）、路由协议（RIP、OSPF、BGP）、广播与多播路由等。
5. 链路层与局域网：涉及链路层功能与服务、差错检测与纠错、多路访问协议、链路层寻址、以太网技术、链路层交换机、PPP协议、链路虚拟化等。

#### 建议：

1. 扎实基础： 确保先修课程的内容熟练掌握，特别是计算机系统结构、程序设计和操作系统的知识。
2. 准备资源：
   1. 阅读推荐教材《Computer Networking-A Top-down Approach》和《计算机网络（第8版）》等，并做好预习。
   2. 配备实验所需的硬件和软件工具，如虚拟机、仿真软件、网络协议分析工具等。
3. 时间管理：课程内容丰富且实验多，需合理安排时间以确保完成各项任务和实验报告。

## 嵌入式系统原理与实践

#### 先修课程：

数字逻辑电路、计算机组成原理、程序设计与C语言、数据结构

#### 课程简介：

《嵌入式系统原理与实践》是一门专业必修课，主要面向计算机科学与技术、电子工程及通信专业的学生。课程涵盖嵌入式系统的基本概念、硬件和软件开发方法，重点培养学生的动手能力。课程内容包括嵌入式微控制器、指令系统、开发工具、外设接口、嵌入式软件设计方法、低功耗设计技术、基本闭环控制、嵌入式操作系统和嵌入式网络协议栈等。

#### 建议：

1. **扎实基础知识**：确保掌握先修课程中的基本概念和原理，这对于理解嵌入式系统的内容至关重要。
2. **提高编程能力**：熟练使用C语言和数据结构，嵌入式系统的开发需要良好的编程技能。
3. **合理安排学习时间**：课程内容广泛且深入，合理安排时间进行预习和复习，确保全面掌握每章的知识点。

## 计算机系统结构

#### 先修课程：

程序设计与c语言、计算机组成原理、操作系统

#### 课程介绍：

计算机系统结构课程是计算机科学和技术专业的学科基础课。课程涵盖现代微处理器和计算机系统的基本架构、设计理念和评估技术，主要内容包括指令集架构、流水线处理器、多级存储结构、Cache优化方法及多处理器系统架构等。

#### 建议：

1. **积极参与课堂讨论和实验：**课堂讨论和实验是理解和掌握课程内容的重要途径，通过实际操作和互动讨论，可以加深对理论知识的理解。
2. **按时完成作业和课堂小测验：**作业和小测验是巩固知识的关键环节，按时完成能够帮助你跟上课程进度，并通过实践进一步理解课程内容。
3. **课外阅读推荐教材和参考资料：**推荐阅读《计算机组成与设计-硬件/软件接口》（MIPS和RISC-V版本），作者：David A. Patterson, John L. Hennessy，以扩展知识面并加深对课程内容的理解。
4. **利用课程内容进行实践：**结合课程所学知识，尝试进行计算机系统相关的小项目或实验，提升实际操作能力，例如设计和优化小型处理器、模拟多级存储结构等。

## 数据库系统原理与实践

#### 先修课程：

数据结构、操作系统

#### 课程简介：

本课程理论课部分系统性地介绍数据库系统的基本概念和相关理论、设计方法及管理技术，内容包括数据库概论、关系模型、SQL和形式化查询语言、数据库设计、事务管理和故障恢复、大数据处理分析、云数据库和分布式数据库系统等。实践课环节主要以MySQL数据库和国产数据库（如华为GaussDB云数据库、人大金仓数据库）为实验环境，介绍数据库的使用与开发技术，涵盖基本操作、数据库管理、数据库编程等内容，通过典型的数据库系统开发案例，介绍数据库应用系统的设计和开发流程。

#### 建议：

1. **基础知识准备：**复习和巩固先修课程（数据结构和操作系统）的知识，特别是数据存储和管理相关内容。
2. **积极参与实践**：实验课涉及的数据库管理系统和开发技术是理解课程内容的关键。通过实验操作掌握MySQL和国产数据库的基本操作和开发技巧。
3. **理论与实践结合：**理论课讲解的数据库设计、事务管理和故障恢复等内容，可以在实践课上通过实际案例进行验证和理解，达到理论与实践结合的效果。
4. **广泛阅读参考资料：**除了课程提供的教材，建议阅读数据库系统相关的经典书籍和最新研究论文，如《数据库系统概念》和《数据库系统实现》等，以拓展知识面和了解数据库技术的发展动态。
5. **关注最新技术：**了解和探索大语言模型（LLM）在数据库领域的应用，通过相关文献和课程内容，掌握大模型赋能数据库技术的基础知识，培养创新思维。

## 编译原理与实践

#### 先修课程：

程序设计语言、离散数学、数据结构、操作系统

#### 课程简介：

《编译原理与实践》是计算机科学与技术专业的专业必修课程，旨在系统讨论编译系统的理论基础、构造方法和实现技术。通过本课程，学生将掌握编译的基本概念、编译器实现的原理和常用方法，了解编译过程及编译系统的构造，具备独立编写一个小型编译系统的能力，从而理解高级程序设计语言的运行机理。

#### 建议：

1. **基础知识掌握：**在修读本课程前，确保已掌握程序设计语言、离散数学、数据结构和操作系统的基本知识。这些内容是编译原理与实践的基础，特别是程序设计语言和数据结构，对于理解编译器的设计和实现至关重要。
2. **理论与实践结合：**本课程要求理论和实践并重。建议学生在学习过程中注重理论知识的理解，并通过编写小型编译器和进行实验操作，加深对所学内容的理解。
3. **使用现代工具：**熟悉并使用Lex和Yacc等编译工具，掌握这些工具的使用方法和原理，有助于更好地理解和实现编译过程中的各个步骤。
4. **多读教材和参考书：**除了课堂讲授内容外，建议多阅读推荐教材和参考书目，扩展知识面，深入理解编译原理。

## 信息工程伦理

#### 先修课程：

计算机导论、人工智能、计算机网络

#### 课程简介：

《信息工程伦理》是一门系统研究工程所引发的社会治理问题以及为工程从业者在职业实践情景中提供价值判断、行为规范和应用方法的学科，属于工程学、社会学与伦理学的交叉学科。课程探讨工程的社会性和实践性、工程风险等，重点关注信息工程师的职业建设、道德推理与分析、责任与权力、职业美德、公正与可靠问题等。通过案例分析，课程旨在帮助学生理解伦理原则并自觉指导未来的信息工程实践​。

#### 建议：

1. **全面掌握先修知识**：确保对计算机导论、人工智能和计算机网络有扎实的理解，这将有助于你在课程中更好地理解和应用伦理原则。
2. **积极参与课堂讨论**：伦理问题往往没有标准答案，通过讨论可以激发你的思维，培养批判性思维能力和伦理决策能力。
3. **案例分析**：多关注课程中的案例分析部分，这些实际案例可以帮助你更好地理解理论知识并应用到实际问题中。
4. **多读相关资料**：虽然课程暂不指定教材，以教学讲义为主，但建议阅读推荐的参考资料，例如李正风、丛杭青等编著的《工程伦理》以及丛杭青讲授的工程伦理MOOC课程。

## 线性代数进阶

#### 先修课程：

线性代数

#### 课程简介：

《线性代数进阶》是一门适合对线性代数和矩阵理论有基本了解的学生的课程。它旨在帮助学生进一步深入理解向量和矩阵的知识，适用于希望在计算机应用、人工智能等领域继续发展的学生。课程特点包括抽象概念较多、要求掌握的处理技术较多以及理论体系较强。主要内容包括：矩阵理论、向量空间理论、矩阵分析。

#### 建议：

1. **扎实掌握基础知识**：确保你对基础线性代数和矩阵运算有良好的理解，这将有助于你更好地跟上课程进度。
2. **积极参与课堂和作业**：课堂讲授为主，辅以适量作业，积极参与可以帮助你更好地理解抽象概念。
3. **参考教材和书目**：建议认真阅读以下教材和参考书目：
   1. 教材：《线性代数》(第三版)，上海交通大学数学系编，北京：科学出版社，2014
   2. 参考书目：

《高等代数与解析几何》(第二版)，陈志杰，北京：高等教育出版社，2008

《高等代数》，北京大学数学力学系，北京：人民教育出版社，1978

《计算机视觉中的数学方法》(第二版)，张贤达，清华大学出版社

1. **关注实际应用**：尝试将所学知识应用到实际问题中，特别是在计算机科学和人工智能领域的应用，这有助于巩固你的理解。

## 计算机基础实践

#### 先修课程：

无

#### 课程简介：

《计算机基础实践》是一门专业选修课。课程内容涵盖计算机日常使用的操作系统、日常办公和应用软件，以及基础网络应用等。该课程以实际操作为主，旨在提高学生的计算机基本操作能力，增进对计算机专业基础知识的了解，培养学生对计算机专业的兴趣，帮助学生实现从理论到应用的平稳过渡。

#### 建议：

1. **积极参与实践：**由于课程以实践为主，积极参与上机操作和实验是掌握技能的关键。
2. **理解并应用知识：**不仅要学习操作技能，还要理解背后的原理，以便在遇到问题时能够灵活应对。
3. **注意日常维护：**通过学习计算机的日常管理和维护方法，养成良好的计算机使用习惯。
4. **培养问题解决能力：**在遇到计算机软硬件问题时，学会分析原因并提出解决方案。
5. **团队合作：**在项目和实训模块中，与同学合作，共同完成任务，培养团队协作能力。

## 面向对象程序设计（基于Java）

#### 先修课程：

程序设计原理

#### 课程简介：

《面向对象程序设计（基于Java）》是一门专业选修课程，旨在教授学生程序设计原理、面向对象程序设计的基本思想、Java语言的基本语法和语义，以及使用Java语言进行程序设计的基本方法。通过本课程的学习，学生将培养使用Java语言进行程序设计和系统开发的能力，为将来学习软件设计开发相关课程及毕业后从事软硬件开发工作打下坚实的基础。课程注重实践，特别强调学生程序设计上机能力的培养。

#### 建议：

1. **理解面向对象的核心概念：**类、对象、继承、封装和多态是面向对象编程的基石，要深入理解它们的含义和应用。
2. **动手实践：**Java编程需要大量的实践，通过编写代码来加深对概念的理解。
3. **关注异常处理：**异常处理是保证程序健壮性的关键，学习如何恰当地使用try、catch和finally等关键字。
4. **深入理解集合框架：**Java的集合框架是日常编程中不可或缺的部分，要熟练掌握各种集合的使用。
5. **阅读和分析代码：**除了自己编写代码，阅读和分析他人的代码同样重要，可以提高编程思维和代码质量。
6. **参与项目实践：**如果有机会，参与实际的软件开发项目，将所学知识应用到实践中，加深理解并提升技能。

## 面向对象程序设计（基于C++）

#### 先修课程：

程序设计原理与C语言

#### 课程简介：

《面向对象程序设计(基于C++)》是一门专业选修课程。本课程旨在通过课堂学习与上机实践，使学生掌握面向对象方法的基本概念以及C++程序设计的原理与方法。课程注重实践，特别是程序设计上机能力的培养，为学生日后学习后续课程和从事应用开发工作，尤其是在面向对象系统开发方面打下坚实的基础。

#### 建议：

1. **理解面向对象的核心概念：**重点掌握类、继承、多态、封装等概念。
2. **注重实践操作：**C++学习需要大量上机实践，通过实践加深理解。
3. **深入理解STL：**熟练掌握标准模板库（STL）的使用，提高编程效率。
4. **关注内存管理：**C++中手动内存管理是常见问题，学习如何有效管理内存。
5. **学习异常处理：**掌握C++的异常处理机制，编写健壮的程序。

## 专业英语

#### 先修课程：

大学英语、计算机导论

#### 课程简介：

《专业英语》是一门面向计算机科学与技术专业三年级本科生的专业选修课，以暑期课程形式开设。课程内容涵盖计算机科学与技术专业英语的听、说、读、写能力培养，专业词汇学习，以及场景式训练。重点在于提升学生在专业领域的英语应用能力，拓展国际化视野，并加强跨文化交流技巧。

#### 建议：

1. **积极参与课堂互动：**专业英语课强调实践，积极参与课堂讨论和角色扮演，提高语言实际应用能力。
2. **广泛阅读英文资料：**利用图书馆和在线资源，广泛阅读英文专业书籍和期刊，增强快速阅读和信息提取能力。
3. **提升科技英语写作技巧：**通过撰写英文摘要和报告，锻炼专业英语写作技巧，注重逻辑性和条理性。
4. **练习英语演讲和汇报：**把握每个公开演讲的机会，练习英语演讲技巧，提高在国际会议中的表达和交流能力。
5. **关注跨文化差异：**了解不同文化背景下的交流方式，培养跨文化沟通意识，为将来的国际合作打下基础。

## 信号与系统

#### 先修课程：

数学分析

#### 课程简介：

《信号与系统》是计算机科学与技术学院提供的专业选修课程，旨在教授信号与系统的描述、分析方法和系统性质。课程内容包括信号的时域和频域分析、傅里叶级数和变换、拉普拉斯和z变换等，培养学生的科学素养和工程实践能力。

#### 建议：

1. **扎实数学基础：**重点复习微积分和复数知识，为理解信号与系统的数学处理打下坚实基础。
2. **积极参与课堂：**通过课堂互动和讨论，深化对信号与系统概念和分析方法的理解。
3. **动手实践**：通过实验和案例分析，将理论知识应用于解决实际工程问题，增强实践能力。
4. **深入学习变换理论：**重点掌握傅里叶变换、拉普拉斯变换和z变换，理解它们在信号分析中的作用。
5. **培养系统思维：**学会从时域和频域多角度观察问题，培养系统化分析和解决问题的思维。