

# 《13182 数据结构》实践考核大纲

## 一、课程性质与目标

### （一）课程性质和特点

数据结构是学习计算机其他专业课程的基础。课程内容从简单数据类型在计算机中表示及存储的方法，到抽象数据类型的定义及在类型上操作的实现过程；从各类线性结构和非线性结构上已有的各种经典算法，到根据实际问题设计并实现高效的算法；从算法的时间、空间复杂度概念，到对算法进行效率评估，可以说，涉及数据及对数据进行操作的方方面面。数据结构是理论与实践相结合的课程，需要用到数学等基础理论知识，也需要使用一种程序设计语言来实现并验证算法，更需要一种逻辑思维模式来设计算法以解决实际问题。通过本课程的学习，同学们能够了解数据结构的基础知识和基本原理，掌握数据结构的存储方式及相关操作的实现，具备利用数据结构进行简单编程处理本专业领域实际问题的初步能力。

### （二）课程目标

设置本课程的主要目的是使同学们了解数据结构的基本概念、基本原理和基本方法，掌握数据结构的逻辑结构和存储结构，实现基本操作；掌握经典算法；具备利用数据结构的基本概念、基本原理和基本方法进行简单编程以处理本专业领域实际问题的初步能力；能够对所实现的算法进行基本的时间复杂度与空间复杂度的评估。

通过本课程的学习，同学们应达到以下目标。

1. 了解数据结构课程的特点，了解该课程在计算机及相关专业课程体系中承上启下的关键作用。掌握数据结构的基本概念、基本原理和基本方法，理解线性结构和非线性结构各自的特点，理解数据结构的逻辑结构与存储结构的定义及相互关系，掌握数据结构的4种基本存储方式。

2. 能够运用数据结构与算法的基本概念，针对具体问题建立抽象数据类型，设计数据的逻辑结构和存储结构。

3. 掌握线性表、树、图的基本概念和基本特性，了解各具体类型的定义、特点和存储方式，掌握不同存储方式下基本操作的实现，并分析各操作实现的效率。掌握线性表、树、图中的经典算法，并能够求解具体问题。

4. 了解排序及查找的基本概念，掌握排序及查找的基本算法、算法特点及适用条件，并能对各排序算法和查找算法进行分析比较。

5. 了解算法的基本概念，掌握实现算法的基本原则，能够使用程序设计语言实现解决具体问题的相关算法。

### （三）课程的重点

本课程的重点内容包括：线性表、树、图的定义、特性及存储方式，不同存储方式下各基本操作的实现；排序和查找算法的特点、实现及比较。

本课程的难点是各数据结构不同存储方式下，相关基本操作的实现，线性结构及非线性结构中经典算法的原理及实现过程，以及解决具体问题的算法的设计与实现。

## 二、考核内容和考核目标

### （一）考核目标

本考试大纲是数据结构课程的个人自学、社会助学和考试命题的依据，本课程的考试范围以本考试大纲所限定的内容为准。

在考核目标方面，本大纲按照识记、领会、简单应用和综合应用四个层次规定同学们应达到的能力层次要求。四个能力层次是递升的关系，后者必须建立在前者的基础上。各能力层次的含义如下。

**识记：**要求同学们能够识别和记忆数据结构课程中有关知识点的概念性内容（如教材中给出的基本定义、基本特点、基本原理、算法特性等），并能够根据考核的不同要求，做出正确的表述、选择和判断。

**领会：**要求同学们在识记的基础上，能够领悟各知识点的内涵和外延，熟悉各知识点之间的区别与联系，能够根据相关知识的特性来解决不同的问题，并能够进行简单的分析。

**简单应用：**要求同学们运用数据结构的少量知识点，分析和解决一般的应用问题，例如分析执行结果、填充程序中的空白、编写简单程序等。

**综合应用：**要求同学们综合运用数据结构的多个知识点，分析解决较复杂的应用问题，并可进行程序设计、分析执行结果、填充程序中的空白、针对应用问题设计算法并编写程序等。

## 第一章绪论

### 一、课程内容

1. 基本概念和术语
2. 算法和算法分析

### 二、学习目的与要求

本章的学习目的是让同学们了解数据结构在计算机应用领域中的作用，掌握数据结构中的基本概念和术语，了解4类基本数据结构，理解逻辑结构与存储结构的含义及相互关系，了解数据结构的4种基本存储方法；掌握抽象数据类型的表示与描述，能够用抽象数据类型表示实际问题；掌握算法的基本概念及重要特性，能够使用类C语言描述算法；掌握算法评估和复杂性度量的基本概念，能够对简单算法进行复杂性评估。

### 三、考核知识点与考核要求

1. 基本概念和术语

识记：数据；数据元素；数据项；数据结构；逻辑结构；存储结构；基本操作；集合；线性结构；树结构；图结构；抽象数据类型。

领会：数据、数据元素、数据项之间的关系；逻辑结构与存储结构之间的关系，存储结构对基本操作实现的影响；数据结构的4种基本存储方法；线性结构与非线性结构的特性。

简单应用：定义抽象数据类型。

## 2. 算法和算法分析

识记：算法及其5个特性；问题规模。

领会：时间复杂度；空间复杂度；大O表示法；增长函数。

简单应用：描述算法；分析程序段的时间复杂度。

## 四、本章的重点、难点

重点：数据结构的逻辑结构、存储结构及相互关系；线性结构与非线性结构；数据结构的4种基本存储方法，算法复杂度的概念及分析方法。

难点：算法复杂度的概念；描述算法；算法复杂度的分析。

## 第二章线性表

### 一、课程内容

1. 线性表的定义和基本操作
2. 线性表的顺序存储及实现
3. 线性表的链式存储及实现
4. 两种基本实现方式的比较
5. 单链表的应用

### 二、学习目的与要求

本章的学习目的是让同学们理解线性表的相关概念，了解其逻辑定义及基本操作，理解线性表数据元素之间的逻辑关系；掌握线性表的顺序存储方式及链式存储方式，领会它们的特点；重点掌握顺序表及链表基本操作的实现，并能进行复杂度分析；针对具体问题灵活选择合适的存储结构，结合运用线性表的基本操作，设计算法解决与线性表相关的实际问题。

### 三、考核知识点与考核要求

#### 1. 线性表的定义和基本操作

识记：线性表；表长；表头元素；表尾元素；直接前驱；直接后继；前驱；有序表；无序表。

领会：线性表的抽象数据类型定义；线性表中元素之间的逻辑关系；线性表中位置的含义；线性表的特点；线性表各基本操作的含义。

简单应用：基于线性表的逻辑定义，给出相关操作结果的逻辑表示。

#### 2. 线性表的顺序存储及实现

识记：顺序表；位置。

领会：顺序存储的基本思想；随机访问的含义；操作位置；表空；表满；表的长度。

简单应用：下标地址计算；顺序表基本操作的实现；顺序表基本操作的前提条件及时间复杂度分析。

综合应用：基于顺序表设计算法来解决具体的应用问题。

### 3. 线性表的链式存储及实现

识记：单链表；表头结点；首结点；表尾结点；终端结点；头结点；表头指针；头指针；单向循环链表；双向链表；双向循环链表。

领会：链式存储的基本思想；头结点的作用；操作位置；顺序访问的含义。

简单应用：链表基本操作的实现及时间复杂度分析；根据问题选择链表类型并实现相关操作；能计算结构性开销。

综合应用：基于链表设计算法来解决具体的应用问题。

### 4. 两种基本实现方式的比较

领会：顺序访问与随机访问的对比；顺序表的使用条件；链表的使用条件。

简单应用：根据问题选择顺序表或链表并实现基本操作。

综合应用：根据问题选择顺序表或链表，设计算法以解决具体的应用问题。

## 四、本章的重点、难点

本章的重点：顺序表及链表的定义、存储特点；顺序表中的地址计算；顺序表及链表基本操作的实现。

本章的难点：运用顺序表和链表的基本操作，设计算法以解决应用问题。。

## 第三章栈和队列

### 一、课程内容

1. 栈
2. 队列
3. 栈和队列的应用

### 二、学习目的与要求

本章的学习目的是让同学们掌握栈和队列的逻辑定义、特点及基本操作，了解它们的逻辑表示方法及使用场景；掌握栈的两种存储方式及各自的特点，掌握这两种存储方式下基本操作的实现及复杂度分析；掌握队列的两种存储方式及各自的特点，掌握这两种存储方式下基本操作的实现，重点掌握循环队列的实现及复杂度分析；了解线性表与栈及队列的关系；灵活运用栈和队列的基本操作，设计算法来解决与此相关的实际问题。

### 三、考核知识点与考核要求

#### 1. 栈

识记：栈；空栈；栈的容量；顺序栈；链式栈；栈顶；栈底；栈顶元素；入

栈；出栈；出栈序列；栈空；栈满。

领会：栈的先进后出特性；能得到的合理出栈序列；顺序和链式实现方式下栈空、栈满的判定条件；顺序和链式实现方式下栈基本操作的定义。

简单应用：能够判定序列是否为栈的合理出栈序列；顺序栈及链式栈基本操作的实现。

综合应用：括号匹配算法的实现；中缀表达式转换为后缀表达式算法的实现；计算后缀表达式算法的实现；设计算法解决栈的应用问题。

## 2. 队列

识记：队列；队头；队尾；队头元素；队尾元素；入队；出队；循环队列；队列空；队列满；队头指针；队尾指针。

领会：循环队列的特点、存储结构及实现机制；循环队列中空、满的含义及判定；链式队列的特点。

简单应用：循环队列和链式队列基本操作的实现。

综合应用：设计算法来解决队列的应用问题。

## 四、本章的重点、难点

本章的重点：栈的概念及合理的出栈序列；顺序栈、链式栈基本操作的实现；循环队列的概念及基本操作的实现；链式队列基本操作的实现；循环队列为空、满的判定。

本章的难点：设计算法来解决栈和队列的应用问题。

# 第四章数组、广义表和字符串

## 一、课程内容

### 1. 数组及广义表

### 2. 字符串

## 二、学习目的与要求

本章的学习目的是让同学们掌握数组、广义表和串的基本概念，掌握数组按行主序和按列主序的存储方式以及二维数组地址计算方法；掌握特殊矩阵的压缩存储方式及相应的地址计算方法；理解广义表的概念，掌握广义表的表示及基本操作；了解模式匹配概念，掌握串的模式匹配算法；能够设计算法解决以数组作为存储结构的简单应用问题。

## 三、考核知识点与考核要求

### 1. 数组

识记：多维数组；首地址；下标；行主序；列主序；映射函数；三元组；三元组表。

领会：二维数组地址计算方法；特殊矩阵的定义、特点及对应的压缩存储方法。

简单应用：构造映射函数；访问采用压缩方式存储的数组中的元素。

综合应用：实现以数组作为存储结构的简单算法。

## 2. 广义表

识记：广义表；表头；表尾；原子（单个元素）；子表；广义表的长度；广义表的深度；递归广义表。

领会：空表；递归表的深度。

简单应用：广义表的基本操作。

## 3. 字符串

识记：字符串；子串；空串；字符串的长度；主串；模式串；比较次数。

领会：字符串模式匹配的含义；特征向量。

简单应用：字符串的基本操作；字符串的特征向量计算；KMP 算法。

## 四、本章的重点、难点

本章的重点：数组行主序、列主序的存储方法；二维数组元素存储地址计算；特殊矩阵压缩存储方法；实现以数组作为存储结构的简单算法；特征向量计算；KMP 算法。

本章的难点：特殊矩阵存储地址计算；实现以数组作为存储结构的简单算法；特征向量计算；KMP 算法。

## 第五章树和二叉树

### 一、课程内容

1. 树的基本概念
2. 二叉树
3. 二叉树的操作
4. 树和森林
5. 哈夫曼树及哈夫曼编码

### 二、学习目的与要求

本章的学习目的是让同学们了解层次结构的概念，了解层次结构与线性结构的不同属性；掌握二叉树的基本概念、基本性质及两种存储方式；能够实现二叉树的基本操作及遍历算法；能够运用二叉树的遍历思想设计算法以解决简单应用问题；理解树及森林的基本概念和存储方式，能够进行树、森林与二叉树之间的相互转换；掌握哈夫曼树及哈夫曼编码的概念，能够针对字符集构造哈夫曼树并设计哈夫曼编码。

### 三、考核知识点与考核要求

#### 1. 树的基本概念

识记：树；子树；结点；边；根结点；双亲结点；兄弟结点；祖先结点；后代结点；孩子结点；层；树的深度；树的高度；结点的度；路径；路径长度；叶结点；分支结点；有序树；森林。

领会：层次结构的含义；树的递归定义；树的表示形式；树中结点个数与边

数的关系。

简单应用：使用不同的方式表示树。

## 2. 二叉树及其操作

识记：二叉树；左子树；右子树；左孩子结点；右孩子结点；满二叉树；完全二叉树；二叉链表；二叉树的遍历；表达式树。

领会：二叉树的递归定义；二叉树的性质；二叉树的顺序存储方式；二叉树顺序存储时保存结点的下标位置；二叉树的链式存储方式；二叉树的遍历过程；二叉树遍历时使用的辅助数据结构；表达式树与表达式的关系。

简单应用：二叉树顺序存储结构与链式存储结构下基本操作的实现；二叉树遍历算法的实现；根据二叉树的遍历序列还原二叉树；表达式树。

综合应用：实现与二叉树相关的算法；相关递归算法的实现。

## 3. 树和森林

识记：树的父结点表示法；树的孩子结点表示法；树的父结点-孩子结点表示法；树的孩子-兄弟表示法；森林；树、森林的遍历。

领会：树、森林之间的关系；树、森林与二叉树的相互转换；树和森林的遍历与二叉树遍历的关系。

简单应用：树、森林遍历的实现；树和森林的其他相关算法的实现。

## 4. 哈夫曼树及哈夫曼编码

识记：原文；译文；定长编码；变长编码；编码长度；前缀特性；编码；译码；字符的权；编码树；哈夫曼编码；哈夫曼树；叶结点的带权路径长度；二叉树的外部带权路径长度（WPL）；带权平均码长。

领会：具有前缀特性的编码方案；哈夫曼编码及哈夫曼树的不唯一性。

简单应用：验证编码方案具有前缀特性；哈夫曼树的构造；哈夫曼编码的生成；译码；树的WPL计算。

综合应用：借助哈夫曼树及哈夫曼编码的思想解决具体问题。

## 四、本章的重点、难点

本章的重点：树、森林及二叉树的基本概念和存储方式；树、森林及二叉树之间的相互转换；二叉树的基本性质；基于二叉链表的递归程序的实现；树、森林及二叉树的遍历；哈夫曼树及哈夫曼编码的构造。

本章的难点：树、森林及二叉树之间的相互转换；基于二叉链表的递归程序的实现；哈夫曼树及哈夫曼编码的构造。

## 第六章图结构

### 一、课程内容

1. 图的基本概念与基本操作
2. 图的存储结构
3. 图的基本操作的实现

4. 图的遍历
5. 图的生成树与图的最小代价生成树
6. 有向无环图及拓扑排序
7. 单源最短路径

## 二、学习目的与要求

本章的学习目的是让同学们掌握图结构的概念、术语及特性，掌握图的邻接矩阵及邻接表存储结构；掌握两种存储方式下图基本操作的实现；掌握图的深度优先搜索和广度优先搜索算法，理解图的连通性及连通分量概念；理解图的生成树概念，掌握求图最小生成树的两个算法；理解有向无环图的概念，掌握图的拓扑排序算法；理解最短路径概念，掌握迪杰斯特拉算法的求解过程；了解各算法的时间复杂度。

## 三、考核知识点与考核要求

### 1. 图的基本概念及基本操作

识记：图；子图；边；顶点；边集；顶点集；关联；稀疏图；密集图；有向边；弧头；弧尾；无向边；有向图；无向图；带权图；邻接点；关联的权；顶点的度；顶点的入度；顶点的出度；完全图；路径；有向路径；简单路径；回路；简单回路；路径长度；有向无环图；连通图；连通分量；强连通图；弱连通图。

领会：顶点与边的关联性；邻接的含义；边与顶点的度的关系；有向路径。

简单应用：计算图中最少边数、最多边数；计算连通图中的边数。

图的存储结构及基本操作的实现

识记：邻接矩阵；邻接表；表结点。

领会：邻接矩阵、邻接表与图的对应关系。

简单应用：根据图的存储结构实现图的基本操作；根据邻接矩阵、邻接表获取图的基本信息。

### 2. 图的遍历

识记：图的遍历；深度优先搜索；广度优先搜索。

领会：图遍历的特性；图的遍历序列；遍历历时使用的辅助数据结构。

简单应用：实现图的遍历算法；基于图遍历思想的算法的实现。

### 3. 图的生成树与图的最小代价生成树

识记：生成树；深度优先生成树；广度优先生成树；最小生成树。

领会：生成树的特性；最小生成树的特性；普里姆算法的求解策略；克鲁斯卡尔算法的求解策略；最小生成树代价的计算。

简单应用：求图的生成树；根据普里姆算法求图的最小生成树；根据克鲁斯卡尔算法求图的最小生成树。

### 4. 有向无环图及拓扑排序

识记：有向无环图；AOV网；前驱；后继；直接前驱；直接后继；活动；拓扑排序；拓扑序列；入度值表。

领会：拓扑排序的过程。

简单应用：判断有向图中是否存在环；求有向无环图的拓扑序列。

#### 5. 单源最短路径

识记：源点；终点；单源最短路径。

领会：迪杰斯特拉算法的求解过程。

简单应用：使用迪杰斯特拉算法求图的单源最短路径。

#### 四、本章的重点、难点

本章的重点：图的基本概念；图的邻接矩阵及邻接表存储结构；两种存储方式下图基本操作的实现；图的两种遍历算法；图的生成树概念及求带权图的最小生成树的算法；有向无环图的概念及图的拓扑排序算法；最短路径的概念及迪杰斯特拉算法。

本章的难点：图的各经典算法的求解过程、算法的实现及时间复杂度分析。

### 第七章内部排序

#### 一、课程内容

1. 排序的基本概念
2. 插入排序
3. 交换排序
4. 选择排序
5. 归并排序
6. 分配排序
7. 有关内部排序算法的比较

#### 二、学习目的与要求

本章的学习目的是让同学们了解排序的基本概念，掌握各排序算法的基本思想、排序过程及其特点，能够实现各排序方法并对各算法进行稳定性和复杂度分析，能够对各排序算法进行比较，理解各排序算法的使用条件。

#### 三、考核知识点与考核要求

##### 1. 排序的基本概念

识记：排序；关键字；主关键字；次关键字；升序；降序；逆序对；内部排序；外排序；稳定排序；交换次数；比较次数。

领会：稳定的含义；评价排序算法的标准；数据的初始排列；排序中的主要操作。

##### 2. 插入排序

识记：直接插入排序；有序子段；待排序元素；希尔排序；增量；增量序列。

领会：直接插入排序的基本思想、特性及复杂度分析，希尔排序方法的基本思想、特性及复杂度分析；组内排序；希尔排序与直接插入排序的相关性。

简单应用：灵活运用直接插入排序和希尔排序方法；能够实现直接插入排序

和希尔排序方法。

### 3. 交换排序

识记：起泡排序；快速排序；枢轴；整体有序；划分；三元取中方法。

领会：起泡排序的基本思想、特性及复杂度分析；快速排序的基本思想、特性及复杂度分析；枢轴的选择方法；枢轴对快速排序效率的影响；分治思想。

简单应用：灵活运用起泡排序和快速排序方法；能够实现起泡排序和快速排序方法。

### 4. 选择排序

识记：简单选择排序；最大堆；大根堆；最小堆；小根堆；堆；堆顶；初始堆；建堆；堆排序。

领会：简单选择排序的基本思想、特性及复杂度分析；堆排序的基本思想、特性及复杂度分析。

简单应用：灵活运用简单选择排序和堆排序方法；能够实现简单选择排序和堆排序方法。

### 5. 归并排序

识记：合并；有序段；归并段；归并排序。

领会：迭代实现的归并排序的基本思想、特性及复杂度分析；递归实现的归并排序的基本思想、特性及复杂度分析。

简单应用：灵活运用归并排序方法；能够实现归并排序方法。

### 6. 分配排序

识记：盒子排序；基数排序；分配；收集。

领会：基数排序的基本思想、特性及复杂度分析。

简单应用：灵活运用基数排序方法；能够实现基数排序方法。

### 7. 有关内部排序算法的比较

领会：各排序方法适用的条件；各排序方法的特性。

简单应用：根据不同条件选择合适的排序方法；从排序结果推测所采用的排序方法。

## 四、本章的重点、难点

本章的重点：排序的基本概念；各排序算法的基本思想、排序过程及其特性；各排序方法的实现及复杂度分析；各种排序算法的比较及使用条件。

本章的难点：各排序算法的实现及特点比较。

## 第八章查找

### 一、课程内容

1. 查找的基本概念
2. 顺序表的查找
3. 树形结构的查找

#### 4. 哈希表及其查找

### 二、学习目的与要求

本章的学习目的是让同学们理解查找的基本概念，掌握顺序查找、折半查找、索引顺序查找方法的基本思想及实现过程，理解各种查找方法的适用条件；理解二叉查找树的概念，掌握二叉查找树操作的定义及实现；理解B树的概念，掌握B树查找及插入操作的定义及实现过程；理解哈希方法中的基本概念，掌握哈希方法；能够对各查找方法进行比较分析。

### 三、考核知识点与考核要求

#### 1. 查找的基本概念

识记：查找；关键字；目标；主关键字；次关键字；单值查找；范围查找；查找表；成功查找；不成功查找；外部查找；内部查找；平均查找长度。

领会：查找算法的适用条件；查找长度的含义。

#### 2. 顺序表的查找

识记：顺序表；顺序查找方法；折半查找方法；折半查找判定树；索引；索引顺序查找；分块查找。

领会：折半查找方法的适用条件。

简单应用：计算顺序查找的平均查找长度；计算折半查找的平均查找长度；分块查找块数和块大小的最优配置。

#### 3. 树形结构的查找

识记：二叉查找树的定义；二叉查找树的特性；二叉查找树的插入及生成；二叉查找树的查找；二叉查找树的删除；查找过程中关键字比较序列；B树。

领会：二叉查找树基本操作的过程；初始数据生成二叉查找树的树形；二叉查找树的树形与查找效率的关系；B树的查找；B树的插入；B树中结点的分裂过程。

简单应用：实现二叉查找树的基本操作；树形查找效率的分析。

#### 4. 哈希表及其查找

识记：哈希表；哈希函数；完美哈希函数；哈希方法；哈希地址；冲突；基本聚集；二次聚集；同义词表；查找成功时的平均查找长度；查找失败时的平均查找长度；装填因子。

领会：直接定址法；平方取中法；折叠法；基数转换法；除留余数法；开放地址法；线性探测法；二次探测法；链地址法；影响哈希方法效率的因素。

简单应用：哈希函数的构造；解决冲突；哈希表的构造。

### 四、本章的重点、难点

本章的重点：顺序查找方法；折半查找方法；分块查找方法；索引查找方法；二叉查找树的插入、删除及查找；B树的查找和插入；哈希方法。

本章的难点：二叉查找树的删除；B树的插入；哈希函数的构造；解决冲突；哈希表的构造。

### 三、参考教材与考核实施要求

#### (一) 本课程使用的参考书

主教材：数据结构，辛运伟、陈朔鹰编著，机械工业出版社，2023年版

辅助教材：(1) Sedgewick R., Wayne K. 著，谢路云译. 算法 (第4版). 人民邮电出版社，2012。(2) 李文辉等. 程序设计导引及在线实践 (第2版). 清华大学出版社，2017。(3) 数据结构习题与实训教程 (C语言描述) (第2版)，王梦菊、齐景嘉等，北京：清华大学出版社，2020。(4) 数据结构教程学习与上机实验指导，李春葆等，北京：清华大学出版社，2020。

#### (二) 本课程的考试要求

1. 考察学生对数据结构基础理论知识的掌握能力，包括数据结构的定义、基本概念（如逻辑结构与物理结构、算法的定义与评价指标），以及线性表、栈、队列、串、数组、树、图等核心数据结构的本质特征、存储方式（顺序存储、链式存储等）及适用场景。

2. 考察学生的基础技术应用能力，包括各类数据结构的基本操作实现（如线性表的增删改查、栈和队列的入栈出栈/入队出队、树的遍历、图的遍历等），以及算法的基本设计与实现思路，能熟练运用编程语言完成基础数据结构操作的编码实现，补全和完善程序代码。

3. 考察学生的综合应用能力，包括根据实际问题选择合适的数据结构与算法、分析算法的时间复杂度与空间复杂度，能将数据结构知识与实际应用结合，解决简单的工程问题（如排序、查找、路径规划等），理解数据结构在程序设计中的核心作用。清晰阐述选择某种数据结构或算法的依据，具备基本的问题拆解与逻辑推导能力。

#### (三) 关于本课程考试命题的若干规定

1. 本门课程采用闭卷考试，时间为150分钟。根据本课程考试所提供的环境条件，携带必要的考试工具（如黑色水性笔）等。

2. 本大纲各章所规定的基本要求，知识点及知识点下的知识细目，都属于考核的内容。考试命题既要覆盖到章，又要避免面面俱到。要注意突出课程的重点、章节重点，加大重点内容的覆盖度。

3. 命题不应有超出大纲中考核知识点范围的题，考核目标不得高于大纲中所规定的相应的最高能力层次要求。命题应着重考核自学者对基本概念、基本知识和基本理论是否了解或掌握，对基本创作实践方法是否会用或熟练。不应出与基本要求不符的偏题或怪题。

4. 本课程在试卷中对不同能力层次要求的分数比例大致为：识记占20%，领会占30%，简单应用占30%，综合应用占20%。

6. 本门课程考试可选用的命题题型范围为单项选择题、读程序写结果题、简

答题、程序填空题、编程题设计题等题型。