



复习与习题

❖ 袁永峰

哈尔滨工业大学



❖ 总成绩（上限100分）

- 平时成绩（50分） + 考试成绩（50分） + 加分（不定）

❖ 平时成绩

- 平时成绩 = 6个实验（36分） + 2个项目（6+8=14分）

❖ 考试成绩

- 期末开卷笔试满分110分，按你实际得分*50%（50分）

❖ 加分（累加到总成绩合计上限5分）

- ACM
- 论坛
- 路径算法+GUI





试题类型



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 单选题
- ❖ 改错题
- ❖ 读程序题
- ❖ 程序设计题
- ❖ 附加题



- ❖ Part 0 计算机的基础知识
- ❖ Part I 程序设计语言的基础知识
- ❖ Part II 数据结构与算法
- ❖ Part III 程序设计语言的实际应用（**GUI、数据库、Web、网络、游戏等等**）

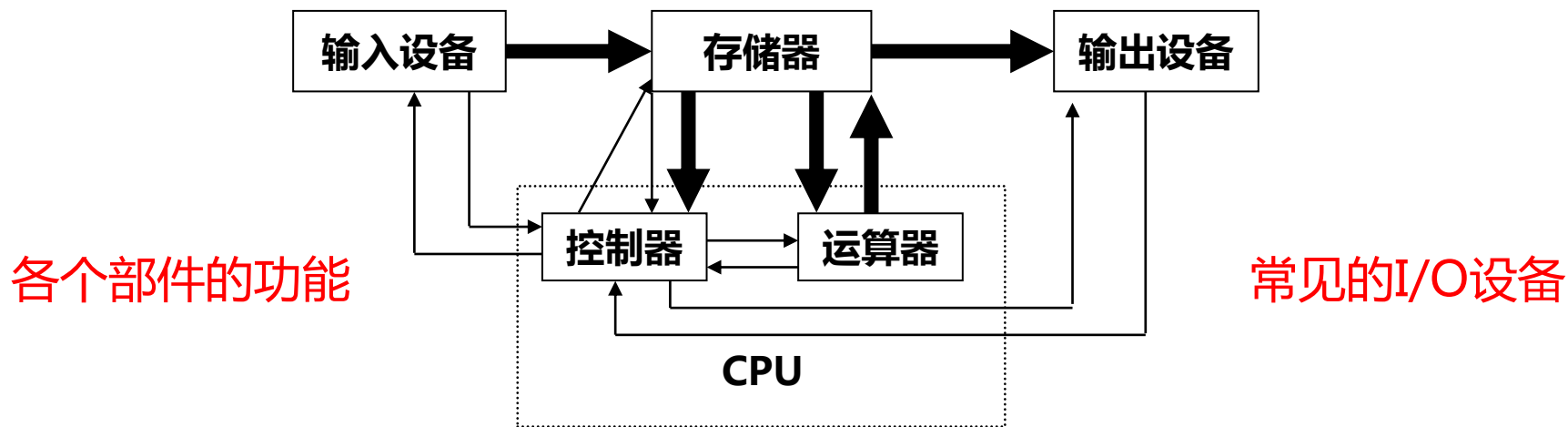
- ❖ 没有讲的重要内容
 - 类与对象（面向对象思维，后继课程Java学习）
 - 内存管理机制（深入了解一门语言时）



0. 计算机的基础知识



- ❖ 计算机，是一种能够按照事先存储的程序自动、高效地对数据进行输入、处理、存储和输出的系统
 - 硬件：计算机的骨架
 - 软件：计算机的灵魂





0. 计算机的基础知识



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ 为什么需要程序设计语言

编码

机器语言
二进制和编码方式

汇编语言
用助记符号编写程序

高级语言
用类似自然语言的
语句编写程序

❖ 为什么不用自然语言与计算机交互？

❖ 程序是一个规定某种计算的指令序列。

❖ 程序 = 数据结构 + 算法

❖ 程序 = 信心：一技之长，实现梦想阶梯

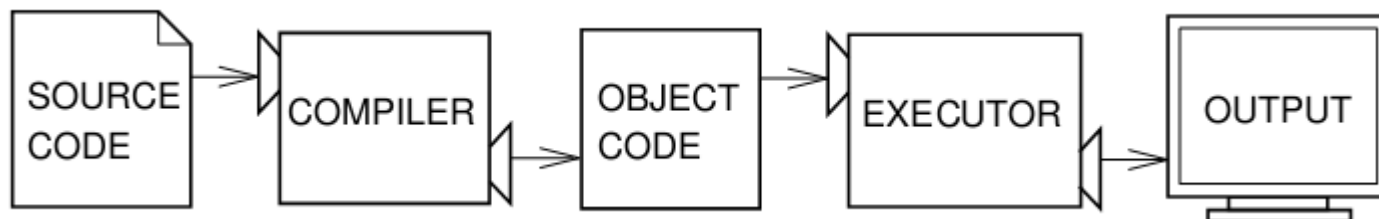


1. 程序设计的基础知识



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ 编译型语言



❖ 解释型语言





1. 程序设计的基础知识



- ❖ 变量、标识符、关键字
- ❖ 基本数据类型、运算符
- ❖ 表达式与优先级
- ❖ 程序结构（顺序、选择（分支）、循环）
- ❖ 函数、参数传递、变量作用域
- ❖ I/O与文件、模块
- ❖ 变量命名规范与程序设计风格
- ❖ 思维：由高到低、由内而外、由简单到复杂（模块化）



- ❖ 用于引用存储在内存中值（数据）的标识符
- ❖ 变量名称 = 值（数值、表达式最后结果、函数返回值）
- ❖ $A=80$
- ❖ $B=1+A*5-2$
- ❖ $C=\text{func}(10,20)$

变量使用前必须初始化



❖ 标识符

- 变量，函数，模块，列表等数据结构的名字

❖ 命名规则

- 可以任意长
- 包含数字和字母、下划线
- 但首个必须是字母或下划线
- 大小写敏感
- 标识符不能是关键字

```
x = 1  
y = 2  
my_name = 'x-man'  
76trombones = 'big parade'  
more@ = 1000000  
class = 'Advanced '
```



关键字



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	class
exec	in	raise	continue	finally
is	return	def	for	lambda
try				

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	



❖ 字符串 (string)

- 使用 ' ' 或 " " 括起来的一系列字符

❖ 整数 (integer)

- 十进制：21，八进制：025，十六进制：0x15

❖ 浮点数 (float)

- 1.65 , 21.0 , 21. , .21 , 2.1E1

❖ 布尔数 (boolean)

- True , False

❖ 复数 (complex)

- 1+1j



❖ 不同类型对象在计算机内表示方式不同

- $5 \rightarrow 101$, $'5' \rightarrow 1001101$

❖ 不同类型对象运算规则不同

- 如：整数的加法和字符串的加法含义不同
- $b = 'a' + 1$



基本类型



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ `int('123')` → 123

❖ `str(123)` → '123'

❖ `float('123')` → 123.0

❖ `float(123)` → 123.0

❖ `bool(123)` → True

❖ `bool(0)` → False



❖ 算术运算符：数值计算

- $+$, $-$, $*$, $/$, $\%$, $**$, $+=$, $-=$

❖ 关系运算符：判断两个值的关系

- $==$, $!=$ ($<>$), $>$, $<$, $>=$, $<=$

- 条件成立 (True) 和条件不成立 (False)

❖ 逻辑运算符：条件的组合关系（短逻辑）

- and , or , not



运算符



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 括号：`()`
- ❖ 一元运算：`+` , `-`
- ❖ 幂次：`**`
- ❖ 算术运算：`*` , `/` , `%` , `//` , `./`
- ❖ 算术运算：`+` , `-`
- ❖ 比较运算：`==` , `!=` , `<>` , `<=` , `>=`
- ❖ 逻辑非：`not`
- ❖ 逻辑与：`and`
- ❖ 逻辑或：`or`
- ❖ 赋值运算：`=` , `*=` , `/=` , `+=` , `-=` , `%=` , `//=`

规则1：

自上而下
括号最高
逻辑最低

规则2：

一元优先
自右向左

规则3：

自左向右
依次结合



- ❖ 自动类型转换
- ❖ 若参与运算的两个对象的类型同，则结果类型不变
- ❖ 若参与运算的两个对象的类型不同，则按照以下规则进行自动类型转换(`bool` \rightarrow `int` \rightarrow `float` \rightarrow `complex`)



练习



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ $x > y$ and $y > z$ 语句等价的是：（ ）

A. $\text{not } x < y$ or $\text{not } y < z$

B. $x > y$ or $\text{not } y < z$

C. $\text{not } x < y$ or $y < z$

D. $x > y > z$

$a = 3^{**}2^{**}2/8$

$\text{print } -7\%2, 7\%-2, -7\%-2$

$b = 5$ or 0

$b = \text{False}$ and 5

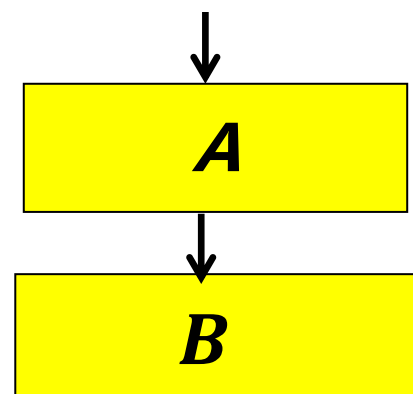
$c = 10-7./2 + 7/2$



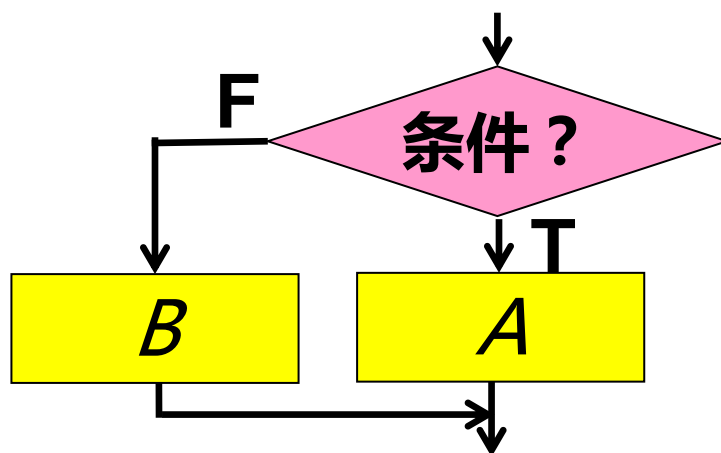
程序的控制结构



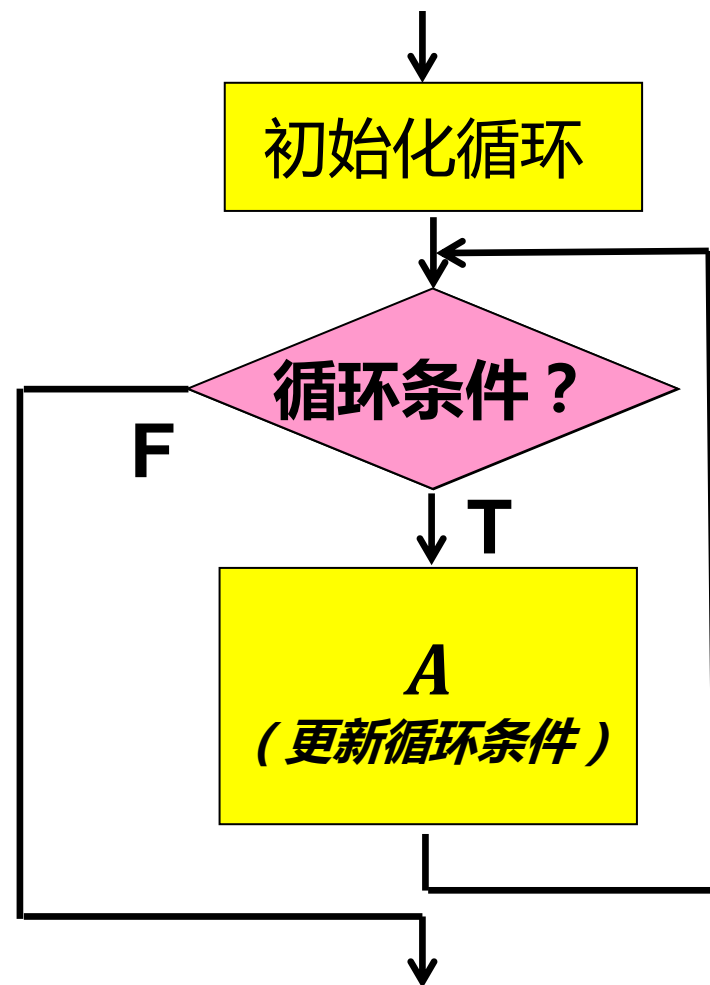
哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY



顺序结构



选择结构



循环结构



❖ 顺序结构

```
x=2  
y=4
```

```
if x > 2:
```

```
    if y > 2:
```

```
        z = x + y
```

```
        print("z is", z)
```

```
    else:
```

```
        print("z is", x)
```

**永远不要忘记
Python中看不见的缩进**

❖ 选择结构

- if
- if....else
- if....elif....elif....else

❖ 循环结构

- for, while , while...e

```
i = initialValue  
while i < endValue:  
    # Loop body  
    ...  
    i = i + 1
```

```
for i in range(initialValue, endValue) :  
    # Loop body
```

❖ 嵌套结构

- 嵌套选择，嵌套循环



- ❖ **break** 强制终止当前循环体
- ❖ **continue** 强制终止当次循环

```
var = 10
while True:
    if var < 9:
        break
    var = var - 9
print var
```

```
var = 10
while True:
    if var < 9:
        continue
    var = var - 9
print var
```

循环体内有return ?



练习



```
y = 0
for i in range(1, 13, 3):
    y += i
print y
```

```
x,y,z = -1,1,-1
if x > 0:
    if y > 0: print 'AAA '
elif z > 0: print 'BBB'
```



❖ 完成特定功能的一个**语句组**，这组语句可以作为一个**单位**使用，并通过函数名执行

关键字

函数名

参数

```
def sum(i1, i2):
```

函数头

缩进

说明文档

```
"""
```

```
    To calculate the sum from i1 to i2  
    i1 is low value, i2 is high value
```

```
"""
```

```
result = 0
```

```
for i in range(i1, i2 + 1):
```

```
    result += i
```

```
return result
```

```
#return sum value
```

函数体

语句



❖ 函数定义

```
def sum(i1, i2):  
    result = 0  
    for i in range(i1, i2 + 1):  
        result += i  
    return result
```

形式参数 (形参)

❖ 函数调用

```
sum(1, 10)
```

实际参数 (实参)



函数调用



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

main()

```
def main():  
    sum(1, 10)  
    sum(20, 37)  
    sum(35, 49)
```

```
def sum(i1, i2):  
    result = 0  
    for i in range(i1,  
i2 + 1):  
        result += i  
    return result
```



函数返回值



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

```
def sum(i1, i2):  
    result = 0  
    for i in range(i1, i2 + 1):  
        result += i  
  
    return result
```

- 函数调用完成后，返回数据
- return语句终止当前函数的执行
- return后的语句将被忽略
- 没有return语句返回None



❖ 局部变量

- 只能在程序的特定部分使用的变量
- 函数内部

❖ 全局变量

- 为整个程序所使用的变量
- 所有函数均可以使用

```
>>> x = 1
>>>
>>> def f1():
...     x = 2
...     print x
...
>>> f1()
2
>>> print x
```

```
>>> x = 1
>>>
>>> def increase():
...     global x
...     x = x + 1
...     print x
...
>>> increase()
2
>>> print x
2
```

```
x = 'a'  
def f():  
    x += 'b'  
    print x ,  
f()  
print x
```

A. ab a

C. b a

B. ab ab

D. 拋出異常

```
lst = []  
def f(lst):  
    return lst[1::2]  
print lst, f([1,3,5,7,9])
```



I/O与文件、模块



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ `raw_input()`和`input()`
- ❖ `print` , 和`print`
- ❖ 转义符 (`\` , `\t`, `\r`, `\n`等)
- ❖ `open('filename' , '[r|w|a|r+|w+|a+]')`
- ❖ `read()`, `readline()`, `readlines()`, `write()`
- ❖ `f.close()`
- ❖ `try: except XXX error:`
- ❖ `import xxx , from xxx import *`



```
radius = raw_input("Enter a value for radius: ")
```

```
area = radius * radius * 3.14159
```

```
f = open( 'f:\\1.txt' )=>f = open( 'f:\\1.txt' , 'r' )
```

```
f.close()
```

❖ 变量命名规范与程序设计风格



2. 数据结构与算法



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 序列型：列表、元组、[字符串]
- ❖ 映射型：字典
- ❖ 组合型：集合
- ❖ 算法与算法设计（IPO）
- ❖ 常用算法：递归、二分、排序（选择、冒泡）
- ❖ 复杂问题求解（数据结构+设计算法）
- ❖ 思维：自顶向下、分而治之、逐步求精



- ❖ **字符串：字符的序列**
- ❖ **列表：一组不同数据类型元素构成的序列**
- ❖ **元组：不可变的序列**

- ❖ **字典：映射关系（键-值）构成的集合**
- ❖ **集合：无序不重复（键）构成的集合**

- ❖ **定义：**
- ❖ **基本操作：**
- ❖ **基本方法：**



可变与不可变



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ 不可变

- 基本数据类型
- 字符串
- 元组
- 固定集合

❖ 可变

- 列表
- 字典
- 集合



基本运算符



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

	字符串	列表	元组	字典	集合
创建					
索引					
切片					
+, *					
in, for					
内建函数					
关系运算					
集合运算					
?= 运算					



基本运算符



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

	字符串	列表	元组	字典	集合
创建	"", ""	[]	(,)	{:, :}	{, }
索引	[]	[]	[]	[key]	X
切片	[:]	[:]	[:]	X	X
+, *	✓	✓	✓	X	X
in, for	✓	✓	✓	✓	✓
内建函数	✓	✓	✓	-	✓
关系运算	✓	✓	✓	✓	✓
集合运算					✓
?= 运算	new	old	new	old	old



索引 (index)



- ❖ 序列中每个元素都有一个索引值 (下标)
- ❖ 索引从0 (前向) 或-1 (后向) 开始

forward index	0	1	2	3	4	5	6	7	8	9	10
	h	e	l	l	o		w	o	r	l	d
backward index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

❖ 索引运算符 []

```
>>> my_str = 'hello world'
```

```
>>> my_str[2]
```

```
'l'
```

```
>>> my_str[-1]
```

```
'd'
```

```
>>> my_str[11]
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
IndexError: string index out of range
```



切片 (Slicing)



❖ 选择子序列

❖ 切法

正序[start : finish]

间隔[start : finish : countBy]

逆序[finish: start : -countBy]

❖ 嵌套

```
lst1 = [1,3,[2,3,4]]
```

```
lst2 = lst1[:]
```

```
lst2[1] = 0
```

```
lst2[2][2] = 0
```

```
print lst1
```



❖ 字典序 (Dictionary order)

- 首先比较两个字符串的第一个字符
- 如果相同，则比较下一个字符
- 如果不同，则字符串的大小关系由这两个字符的关系决定
- 如果其中一个字符为空（较短），则其更小

❖ 内建函数

- `len ()`、`sum ()`、`max ()`
- `sorted ()`



❖ 课件和编程常用的

❖ 注意细节和区别

- `lst.append()` 和 `lst.extend()`
- `string.find()` 和 `string.index()`

❖ 实践、实践、再实践



❖ 算法 (Algorithm)

- 一个计算的具体步骤，常用于计算、数据处理和自动推理

❖ Donald Knuth对算法下的定义：

- 输入：一个算法必须有零个或以上输入量
- 输出：一个算法应有一个或以上输出量，输出量是算法计算的结果
- 明确性：算法的描述必须无歧义
- 有限性：一个算法是能够被任何图灵完备系统模拟的一串运算
- 有效性：能够实现，算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现



- ❖ 量化一个算法的运行时间为输入长度的函数
- ❖ 随着 n 的增大， n 和算法性能之间的关系称为**算法的计算复杂度 (Computational Complexity)**
- ❖ **时间复杂度 (Time Complexity) 和空间复杂度 (Space Complexity)**
- ❖ 算法的时间复杂度是算法的时间量度，是指执行算法所需要的计算工作量



简单算法（二分、排序）



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

```
def bi_search(lst, v):
    low = 0
    up = len(lst) - 1
    while low < up:
        mid = (low + up) // 2
        if lst[mid] < v:
            low = mid + 1
        elif lst[mid] > v:
            up = mid
        else:
            return mid
    return -1

def selection_sort(lst):
    for i in range(len(lst) - 1):
        min_idx = i
        for j in range(i + 1, len(lst)):
            if lst[j] < lst[min_idx]:
                min_idx = j
        swap(lst, min_idx, i)

re_lst = [42, 16, 84, 12, 77, 26, 53]
selection_sort(re_lst)
print re_lst
```



简单算法（二分、排序）



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

```
1 def swap(lst, a, b): // 交换列表中a,b 对应位置的两个元素
2     tmp = lst[a]
3     lst[a] = lst[b]
4     lst[b] = tmp
5
6 def bubble_sortv3(lst):
7     exchanged = True // 是否进行过交换操作的标志
8     top = len(lst) - 1
9     while exchanged:
10         exchanged = False
11         for i in range(top):
12             if lst[i] > lst[i+1]:
13                 swap(lst, i, i+1)
14                 exchanged = True
15         top -= 1
```



递归



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

```
def p(n):
```

```
    if n == 1 or n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * p(n-1)
```

递归终止

递归条件

```
n = input("请输入一个整数:")
```

```
print n, " !的值为 :", p(n)
```

在函数定义有直接或间接调用自身



❖ 优势 (strength)

- 能使一个蕴含递归关系且结构复杂的程序简洁精炼，增加可读性
- 特别是在难于找到从边界到解的全过程的情况下，如果把问题推进一步,其结果仍维持原问题的关系

❖ 劣势 (weakness)

- 嵌套层次深，函数调用开销大
- 重复计算



递归



```
from turtle import forward, backward, left, right

def branch(length, level):
    if level <= 0:
        return
    forward(length)
    left(45)
    branch(0.6 * length, level - 1)
    right(90)
    branch(0.6 * length, level - 1)
    left(45)
    backward(length)
left(90)
branch(100, 4)
```



3. Python的应用



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ GUI（控件、事件处理）
- ❖ 数据库（基本概念、数据表、SQL语句）
- ❖ Web（安装、部署）
- ❖ 思维：系统（整体性与架构）、单元与集成、测试



- ❖ 某些英语单词的字母经过重新排列后，能获得另外一个单词，如可以将“cinema”转换成另一个单词“iceman”。编写程序，将词表中由相同字母组成的单词聚成一类，并按照单词个数由多到少的顺序输出各类中的全部单词（每类占一行，单词按字典序由小到大排列，之间用空格分隔），若每类中单词个数相同，则按每类中第一个单词的字典序输出各个类别。其中，单词表存放在“words.dic”文件中。



祝大家取得好成绩

每天都在成长

谢谢