

# 阶段总结

---

- **10次课**，已经讲完了基本程序设计的主要内容，以及“什么是计算”的一部分内容
- **编程序**，就是指挥计算机完成我们想做的计算
  - 问题在前，首先是要弄清需要做的事情
  - 例如，看懂作业题，弄清题目要求做什么（有问题吗？）
- **例**：要求定义一个函数，完成某种计算
  - 可能还要求做几个例子
  - 什么是完成了作业？
- **注意区分函数的结果（返回值）和程序的输出**

# 课程内容概要

---

- 用计算解决问题的过程和工作步骤
- 编写程序的过程：编程、测试、更正和改进
- 程序计算的基本形式（及其形成的计算过程）
  - 直线型结构
  - 分支结构
  - 循环结构
  - 递归定义的函数
- 变量和变量的值（数据，对象）
- 函数抽象：定义和使用
- 类型，在计算中的作用
- 编码和数制转换，二进制/十进制/八进制/十六进制

# 课程内容概要

---

## ■ Python 语言的基本编程机制

- 基本类型: **int, float, str, bool**

- 运算符和表达式

  - 优先级, 结合关系, 括号, 计算的顺序

- 基本语句:

  - 赋值语句, 函数调用, 等

  - assert, global/nonlocal** 声明, **import**, 等

- 计算流程控制语句:

  - 复合语句: **if, while, for**

  - 简单控制语句: **break, continue, return**

- 函数的定义和调用

# 课程内容概要

---

## ■ 概念

- 语法和语义
- 类型和对象
- 表达式和求值
- 优先级，结合顺序
- 程序错误（语法/类型/逻辑）
- 精确和近似的计算
- 标识符和关键字
- 标准库
- 变量和赋值
- 扩充赋值运算符
- 断言

- 输入和输出

- 循环和递归

- 函数定义和调用

  - 形参（形式参数）和实参  
（实际参数）

  - 函数返回值（结果）

- 迭代器

- 终止性

- 作用域，局部，全局，外围

- 环境和状态

- 模块，执行和导入

- 匿名函数和 **lambda** 表达式

# 课程内容概要

---

## ■ 技术:

- 浮点数的使用和问题
- 检查参数（类型和值）
- 断言的使用和意义
- 典型循环模式
- 函数的局部定义
- 给程序计时
- 函数的函数参数
- 计算框架和函数功能分解
- **lambda** 及其使用（匿名函数，作为返回值）
- 随机数和计算机模拟

# 关键字

---

## ■ 已经讨论过的关键字（24/33）

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

## ■ 分类：

- 特殊字面量 **None, True, False**
- 运算符 **and, or, not, in**（in 还用在 for 头部）
- 基本语句 **assert, break, continue, pass, return, del**, 模块导入 **from-import**, 变量声明 **global** 和 **nonlocal**
- 控制语句 **for-in, while, if-elif-else**
- 函数定义 **def**, 描述函数的特殊表达式 **lambda**

# 期中考试题目的可能类型

---

- 填空题
- 正误题
- 单选和多选题（多选题分项记分）
- 简答题
- 读程序题
- 编程题（在答卷纸上写出程序，请尽可能整齐，以便评判。可以根据需要写注释或另外说明）