哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

星际穿越

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

学编程如同Interstellar一样，
只有跳进“黑洞”，才能了解
它的秘密

but…

完成实验所需的编程技能，只有30%来自于课堂，其他的需要靠你自学

so …?

完成第x周的实验所需的知识，可能在第x+n周才能在课堂上出现

so …?

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

只能用Java，但我不会/不喜欢Java

so…?

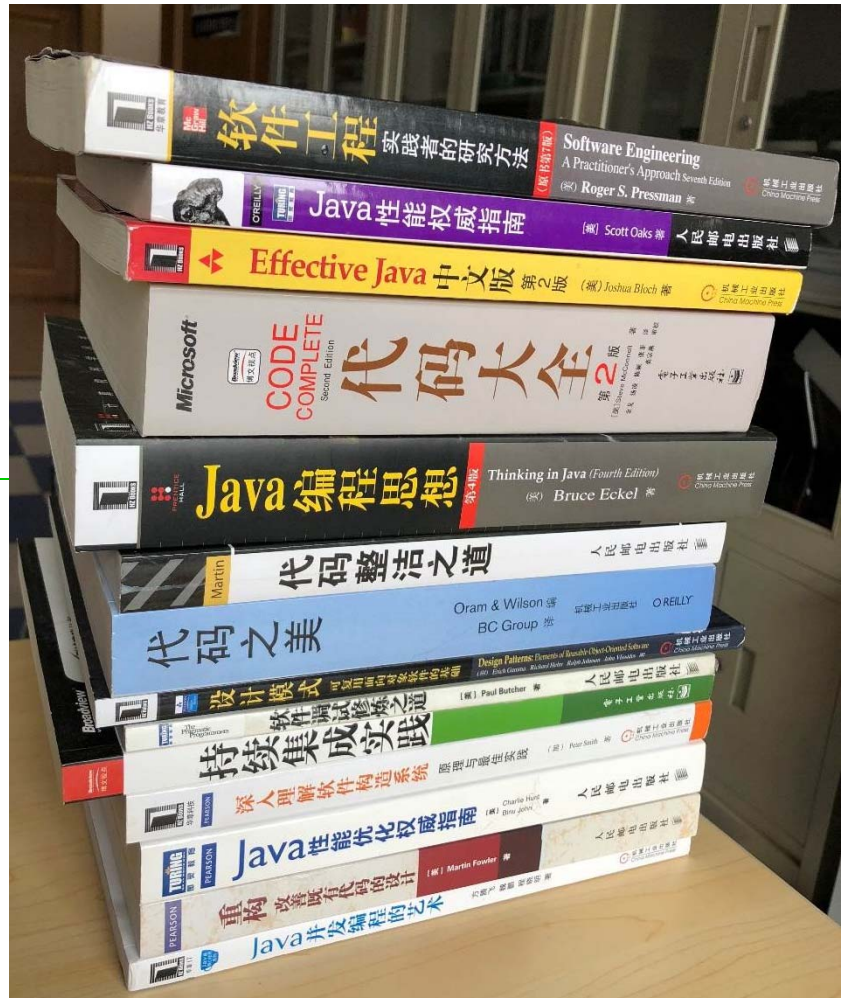
哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

无教材…so?

无教材

但有10+本参考书

…so?



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

MIT 6.031 + CMU 15-214 so?

Rank	School name	Score
#1 Tie	Carnegie Mellon University Pittsburgh, PA	5.0
#1 Tie	Massachusetts Institute of Technology Cambridge, MA	5.0
#1 Tie	Stanford University Stanford, CA	5.0
#1 Tie	University of California—Berkeley Berkeley, CA	5.0

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

英文的讲义
so?

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

5学分
80学时
延续17周
闭卷考试
no cheat sheet…
so?

JAVA SYNTAX CHEAT SHEET

Control Flow

Selection
If
Else
Switch
Case

Loop
While
Do
For

Exception
Throw
Try
Catch
Finally

Branch
Return
Break
Continue
Label

BASIC DEFINITIONS

Class	Describes a particular kind of object. It can contain related methods and variables.
Method	A function defined in a class. Methods implement the behavior for objects.
Object	The principal building blocks of Java. Objects consist of variables (data) and methods (functionality).

COMMENTS

HTML Comments

<!-- comment -->	Sent to the client in the viewable page source.

JSP Commens (Not Sent to Client)

<%-- comment --%>	Comments in JSP file.
// comment	Comment in scriplet part of JSP file.

ELEMENTS

Declaration

<%! declaration %>	Creates a global variable or method.

Expression

<%= expression %>	Statements evaluated on the server before the page is outputted to the client.

Page Directive

<%@ directive %>	Attributes that apply to the entire page.

Scriplet

<% code fragment of one or more lines %>	Contains a block of scripting code which is executed when the page is generated.

Taglib Directive

<%@ taglib uri="URIToTagLibrary" prefix="tagPrefix" %>	Defines a tag library and prefix for tags used in a JSP page.

SEPARATORS

( )	Used to surround parameters
{ }	Defines a block of code for a class or method or to contain the values of automatically initialized arrays
[ ]	Declares arrays or references array values
;	Denotes the end of a statement
,	Separates variables
.	Separates package names from subpackages/ classes or a variable/method from a reference variable

PRIMITIVE DATA TYPES

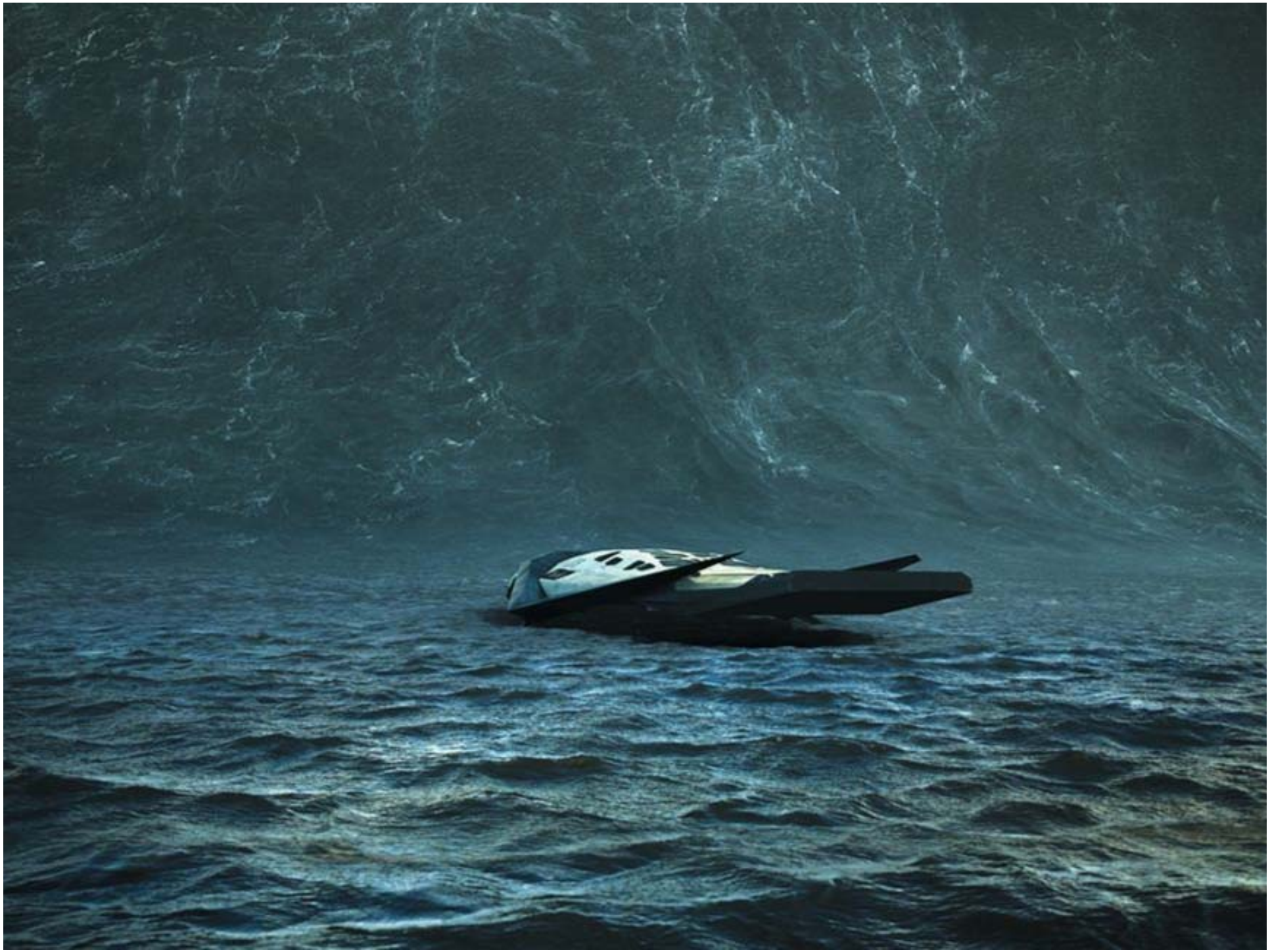
Type	Description	Bits
	(Integers)	
byte	Byte-length integer	8
short	Short integer	16
int	Integer	32
long	Long integer	64
	(Real Numbers)	
float	Single-precision floating point	32
double	Double-precision floating point	64
	(Other)	
char	A single character	16
boolean	A boolean value (true or false)	1

OPERATORS

+	addition of numbers, concatenation of Strings
+=	add and assign numbers, concatenate and assign Strings
-	subtraction
-=	subtract and assign
*	multiplication
*=	multiply and assign
/	division
/=	divide and assign
%	take remainder
%=	take remainder and assign
++	increment by one
--	decrement by one
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
!	boolean NOT
!=	not equal to
&&	boolean AND
||	boolean OR
==	boolean equals
=	assignment

Key Words

abstract
assert
boolean
break
byte
case
catch
char
class
const
continue
default
do
double
else
enum
extends
false
final
finally
float
for
goto
if
implements
import
instanceof
int
interface
long
native
new
null
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
true
try
void
volatile
while



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Tips:

写更多的代码
把使用工具变成习惯
独立完成实验
遇到问题，尽可能独立解决
多与教师/TA交流
多总结多反思/写博客

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

软件构造：开发高质量的软件系统

Software Construction: Developing High-Quality Software Systems

Wang Zhongjie
rainy@hit.edu.cn

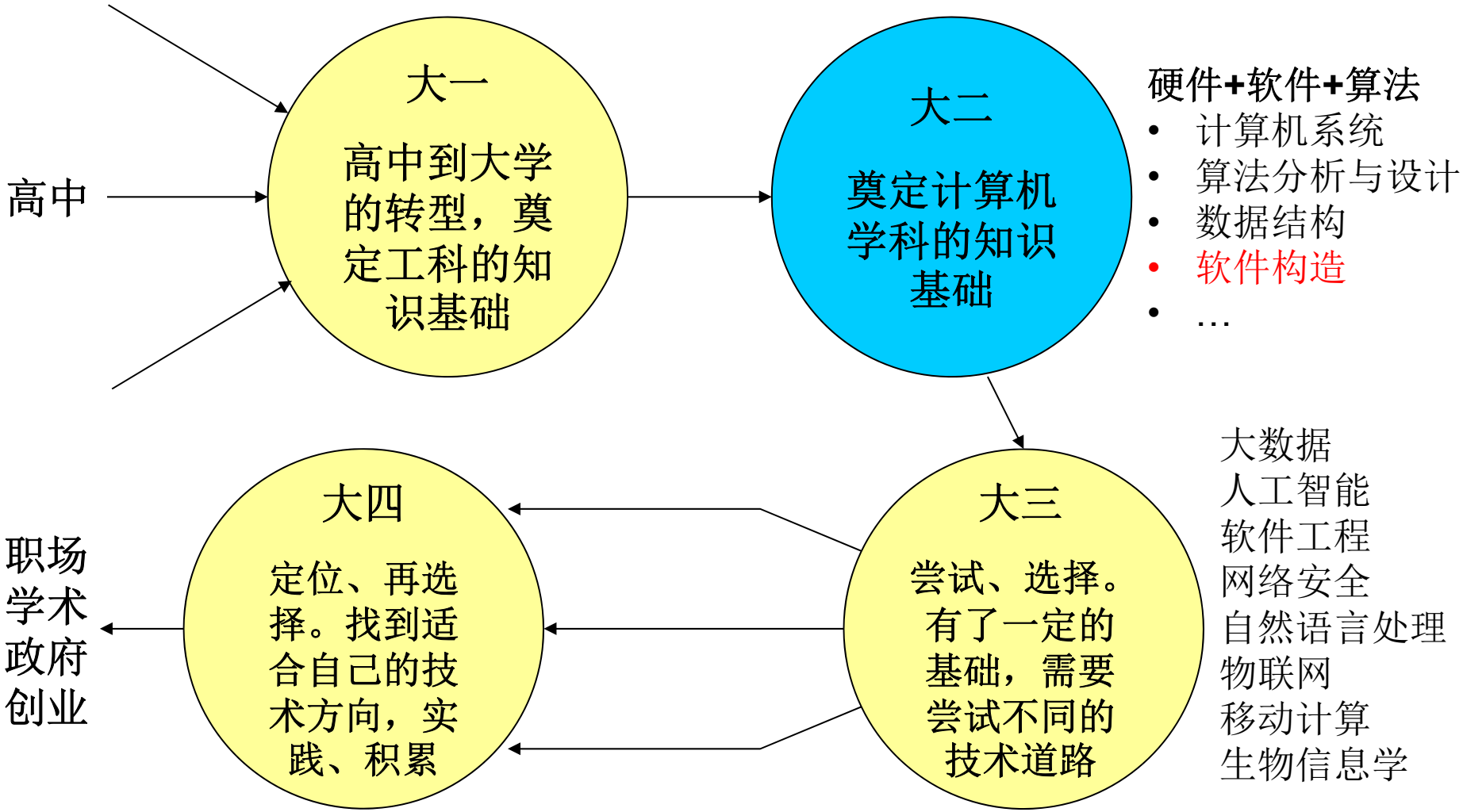
February 26, 2018

任课教师

- **王忠杰 教授，博士生导师**
  - 企业与服务智能计算研究中心 (ICES)
  - 电子邮件： rainy@hit.edu.cn
    rainy.wang@gmail.com
  - 联系电话： 18604507162
  - 地点： 新技术楼510房间
  - 答疑时间： 每周三10:00-11:30 (1-15周)
  - 研究方向：
    - “云+端”融合的服务计算
    - 软件体系结构
    - 实证软件工程

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

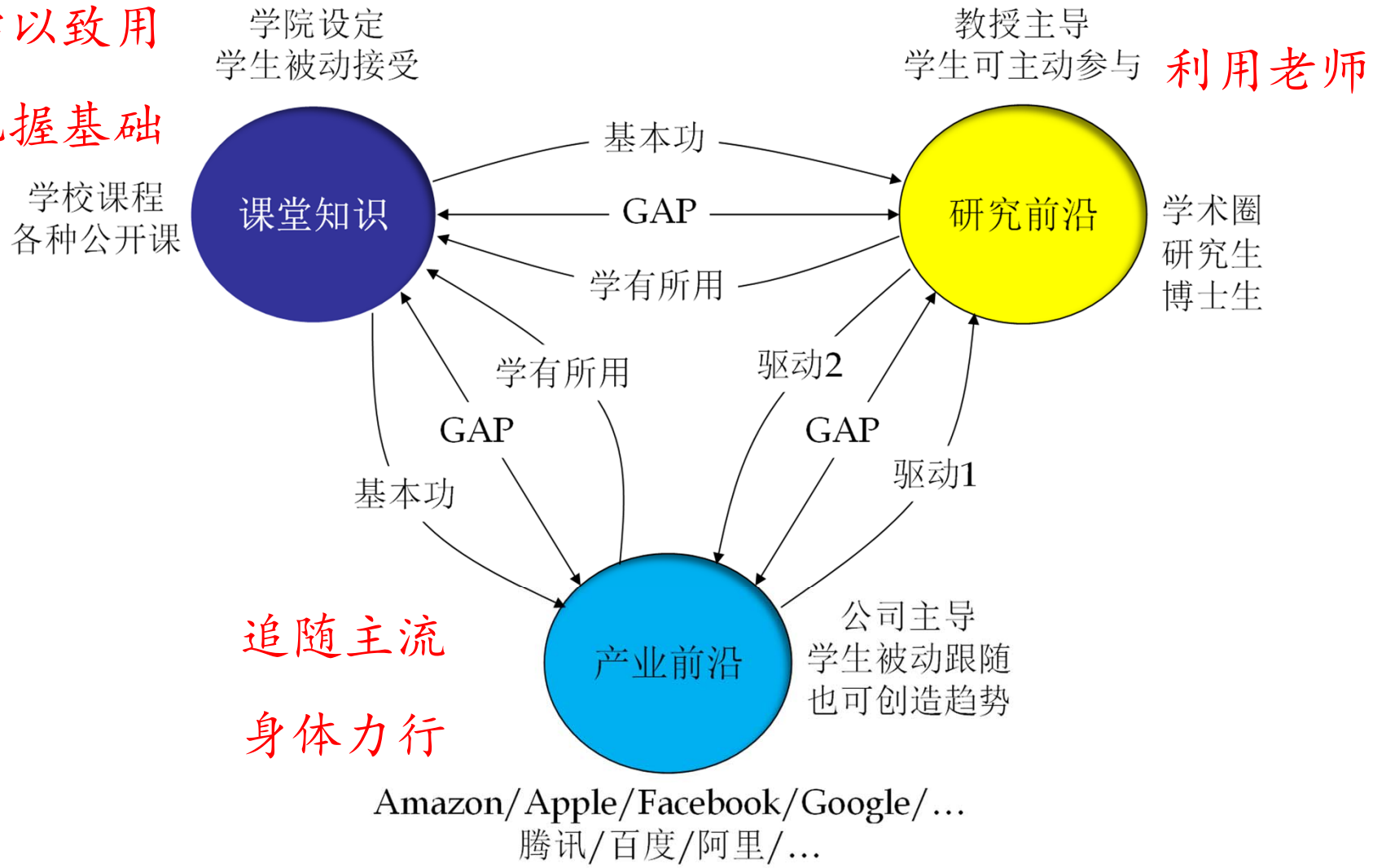
建议 1：打好基础，为“选择”做准备



建议2：弥补课堂和现实的gap

学以致用

把握基础



Goals of this Course

- **Goal: understanding both the building blocks and the design principles for construction of software systems** 构造原理？
  - 在高级语言程序设计的基础上，认识软件构造的质量标准与目标，学习软件构造的基本过程，从而具备面向质量目标的复杂软件构造方法与能力
  - 深入学习抽象数据类型 ADT 与面向对象编程 OOP
  - 初步掌握面向关键质量目标（可理解性、可维护性、可复用性、健壮性、时空性能）的软件构造基本技术
  - 了解软件代码重构和面向更复杂软件系统的高级构造技术
- **For each desired program behavior there are infinitely many programs** 多种不同的软件构造方案，有什么差异？如何选择？
  - What are the differences between the variants?
  - Which variant should we choose?
  - How can we synthesize a variant with desired properties?

Goals of this Course

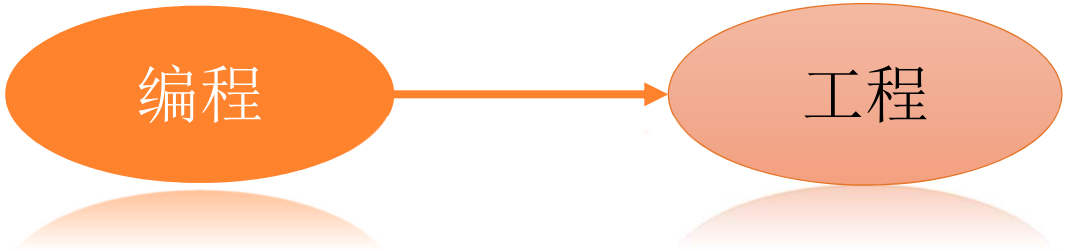
程序设计与实现能力

- 了解软件开发过程中应考虑哪些质量目标
- 掌握面向关键质量目标的软件基本构造技术
- 形成面向质量目标的软件开发思维模式

系统设计与实现能力

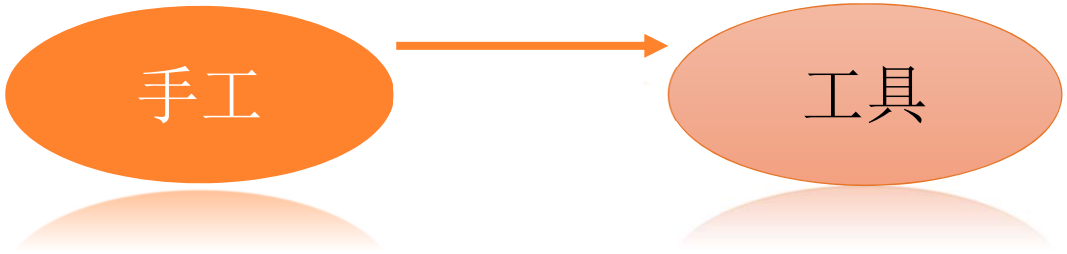
- 掌握“面向抽象编程”的核心思想和面向对象软件开发的基本过程
- 能够对实际应用问题进行抽象和建模
- 利用模型与开发者和用户进行有效表达和沟通

Goals of this Course



系统分析与评价能力

- 从关注单一开发环节到关注全开发过程的转换
- 根据用户期望质量特性进行全生命周期**系统分析与评价**
- 发现系统设计的缺陷并做出优化和改进



利用现代软件构造工具的能力

- 了解复杂软件系统相对于简单程序的本质差异
- 初步掌握利用各类软件开发工具进行编码、测试和质量保障
- **利用现代软件构造工具**进行高质量和高效率软件开发

A typical software design process

1. **Discuss software that needs to be written**
2. **Write some code**
3. **Test the code to identify the defects**
4. **Debug to find causes of defects**
5. **Fix the defects**
6. **If not done, return to step 1**

写代码----试错----改错，如此循环

Better software design

- **Think before coding** 未雨绸缪
- **Consider non-functional quality attributes** 考虑非功能质量属性
  - **Maintainability, extensibility, performance, …**
- **Propose, consider design alternatives** 考虑多种设计选择
- **Make explicit design decisions** 把设计决策明确写下来

- **Using a design process…**
  - A design process organizes your work
  - A design process structures your understanding
  - A design process facilitates communication

Design goals, principles, and patterns

- **Design goals** enable evaluation of designs
  - e.g. maintainability, reusability, scalability
- **Design principles** are heuristics that describe best practices
  - e.g. high correspondence to real-world concepts
- **Design patterns** codify repeated experiences, common solutions
  - e.g. template method pattern

设计目标：编程的“视野”

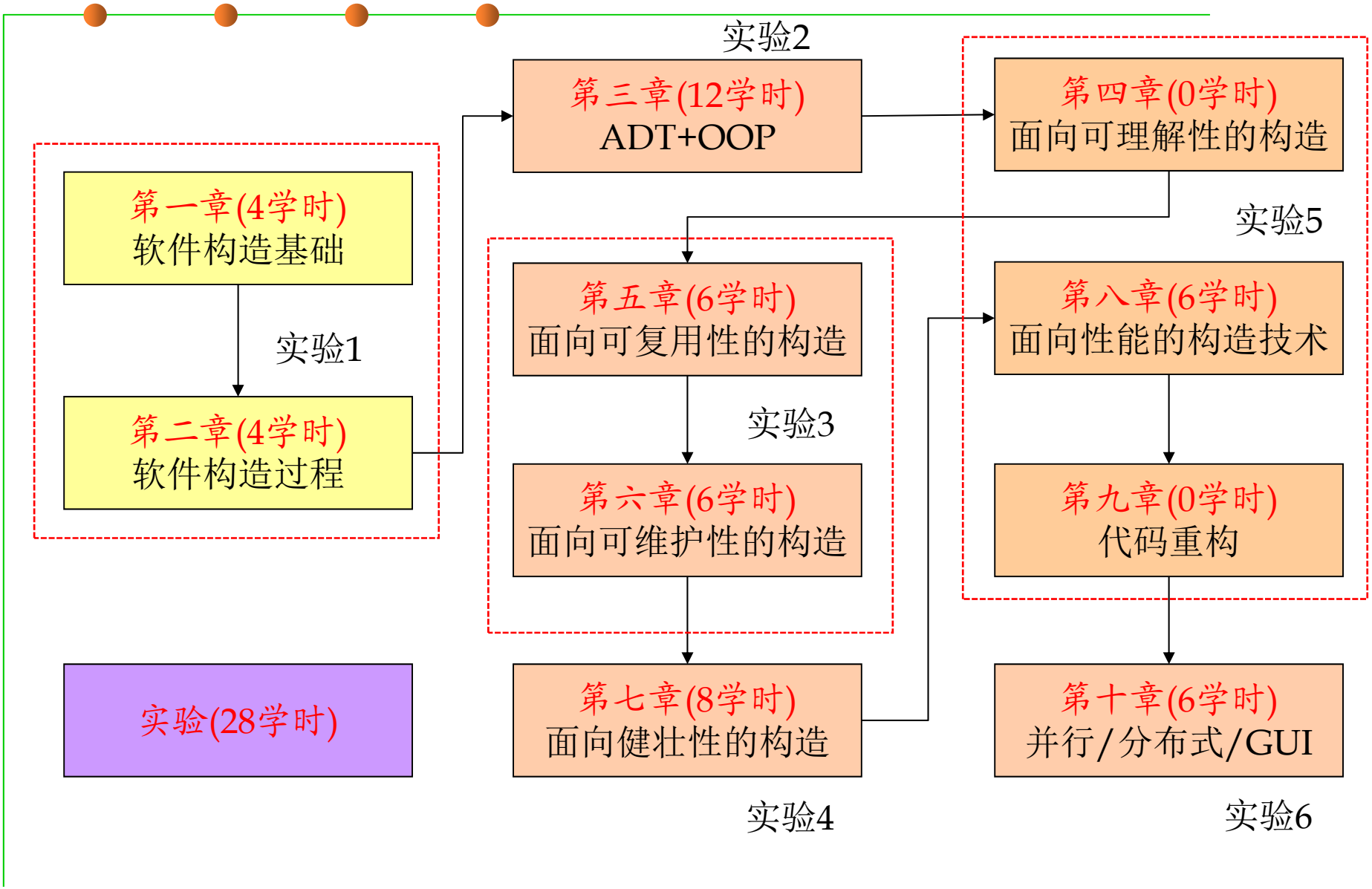
设计原则：编程的“标尺”

设计模式：编程的“经验”

Learning goals

- Ability to **design** medium-scale programs
- Understanding **OO programming** concepts & design decisions
- Proficiency with basic **quality assurance** techniques for functional correctness
- Fundamentals of **concurrency and distributed systems**
- Practical skills

课程章节安排


第一章(4学时)
软件构造基础
实验1
第二章(4学时)
软件构造过程
实验(28学时)
实验2
第三章(12学时)
ADT+OOP
第五章(6学时)
面向可复用性的构造
实验3
第六章(6学时)
面向可维护性的构造
第七章(8学时)
面向健壮性的构造
实验4
第四章(0学时)
面向可理解性的构造
实验5
第八章(6学时)
面向性能的构造技术
第九章(0学时)
代码重构
第十章(6学时)
并行/分布式/GUI
实验6

课程简介

- **授课对象:** 计算机学院2016级本科生
- **课程分类:** 核心基础课
- **学时:** 80 (52+28)
- **先修课程:** C/C++/Java高级语言程序设计；
  计算机系统；数据结构与算法；
- **上课时间/地点:**
  - 1-14周　周一/周三1-2节　正心楼41
  - 1-15周　周一5-6节　格物楼机房213/214
- **考试时间:**
  - 17周　周日10:00-12:00　正心楼23、24

课程网站

- 软件构造

  http://cms.hit.edu.cn/course/view.php?id=184

- 课件、实验与作业要求、各类通知均在此网站发布，实验/作业完成之后也需通过此网站提交。
- 可通过CMS网站提出问题，与教师和同学进行交流。

推荐学习资料

- **MIT Course 6.031: Software Construction**

  http://web.mit.edu/6.031/www/sp17/

6.031: Software Construction

Spring 2017 · Course Staff · MWF11-12:30 (34-101)

General

General Information
Collaboration and Public Sharing
Code Reviewing
Nanoquiz Grading and Makeups
I have a question, who do I ask?

Calendar: classes, assignments, OH/lab

Getting Started

Getting Started: Java, Eclipse, & Git
Getting Started: Java Tutor
Getting Started: Eclipse FAQ

Problem Sets

0: Turtle Graphics
1: Around the World
2: Poetic Walks
3: Calculus
4: Multiplayer Minesweeper

Project

Phase 1: Norn Mailing List System
Phase 2: WebNorn Mailing List System

Quizzes

Quiz 1 and Quiz 1 solutions
Quiz 2 and Quiz 2 solutions

Quiz Archive

Course Archive

Previous semesters

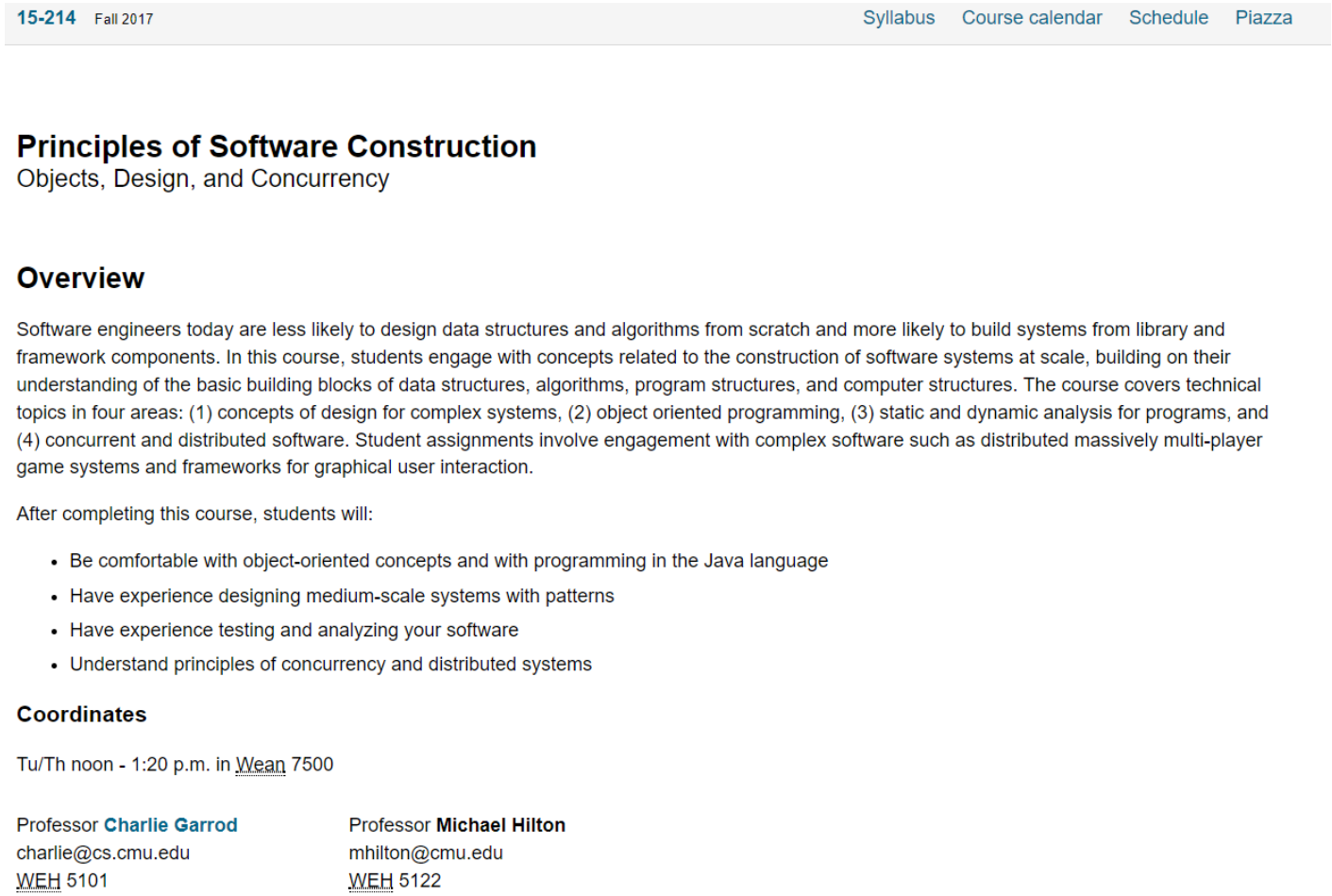
Readings

01: Static Checking
02: Basic Java
03: Testing
04: Code Review
05: Version Control
06: Specifications
07: Designing Specifications
08: Avoiding Debugging
09: Mutability & Immutability
10: Recursion
11: Debugging
12: Abstract Data Types
13: Abstraction Functions & Rep Invariants
14: Interfaces & Enumerations
15: Equality
16: Recursive Data Types
17: Regular Expressions & Grammars
18: Parsers
19: Concurrency
20: Thread Safety
21: Locks & Synchronization
22: Queues & Message-Passing
23: Team Version Control I
24: Sockets & Networking
25: Callbacks
26: Map, Filter, Reduce
27: Little Languages I
28: Little Languages II

推荐学习资料

http://www.cs.cmu.edu/~charlie/courses/15-214/2017-fall

- **CMU 15-214 Principles of Software Construction: Objects, Design, and Concurrency**



15-214 Fall 2017 Syllabus Course calendar Schedule Piazza

Principles of Software Construction
Objects, Design, and Concurrency

Overview

Software engineers today are less likely to design data structures and algorithms from scratch and more likely to build systems from library and framework components. In this course, students engage with concepts related to the construction of software systems at scale, building on their understanding of the basic building blocks of data structures, algorithms, program structures, and computer structures. The course covers technical topics in four areas: (1) concepts of design for complex systems, (2) object oriented programming, (3) static and dynamic analysis for programs, and (4) concurrent and distributed software. Student assignments involve engagement with complex software such as distributed massively multi-player game systems and frameworks for graphical user interaction.

After completing this course, students will:

- Be comfortable with object-oriented concepts and with programming in the Java language
- Have experience designing medium-scale systems with patterns
- Have experience testing and analyzing your software
- Understand principles of concurrency and distributed systems

Coordinates

Tu/Th noon - 1:20 p.m. in Wean 7500

Professor **Charlie Garrod**
charlie@cs.cmu.edu
WEH 5101

Professor **Michael Hilton**
mhilton@cmu.edu
WEH 5122

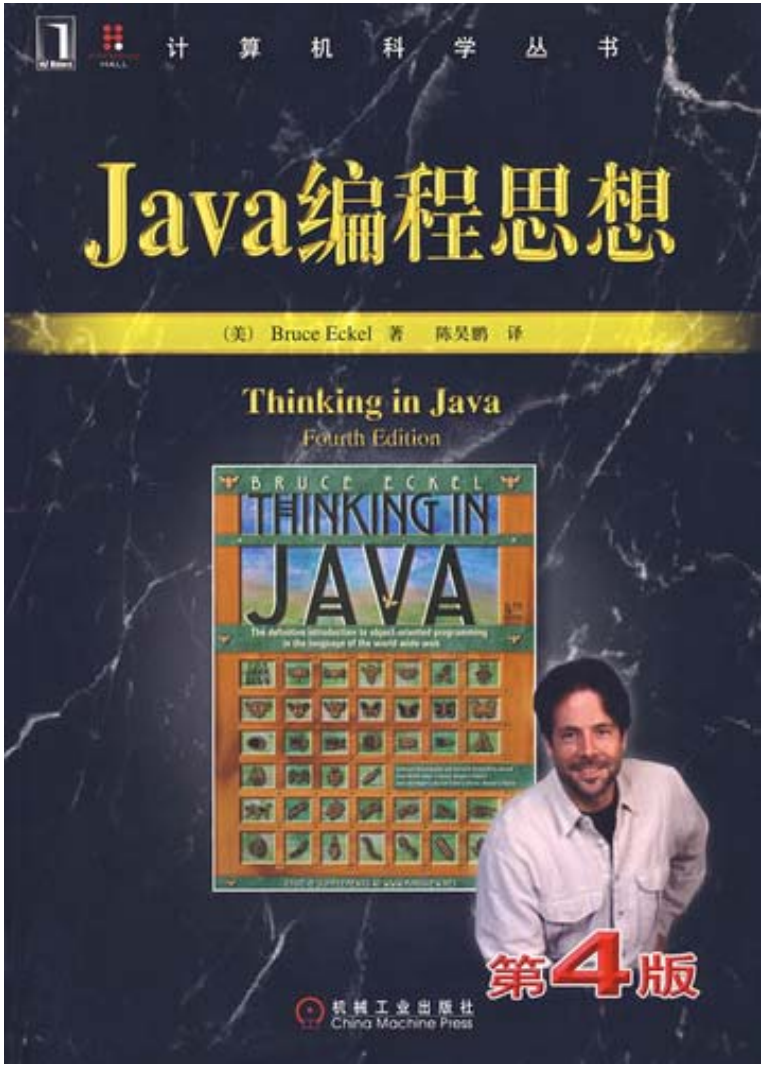
推荐阅读材料

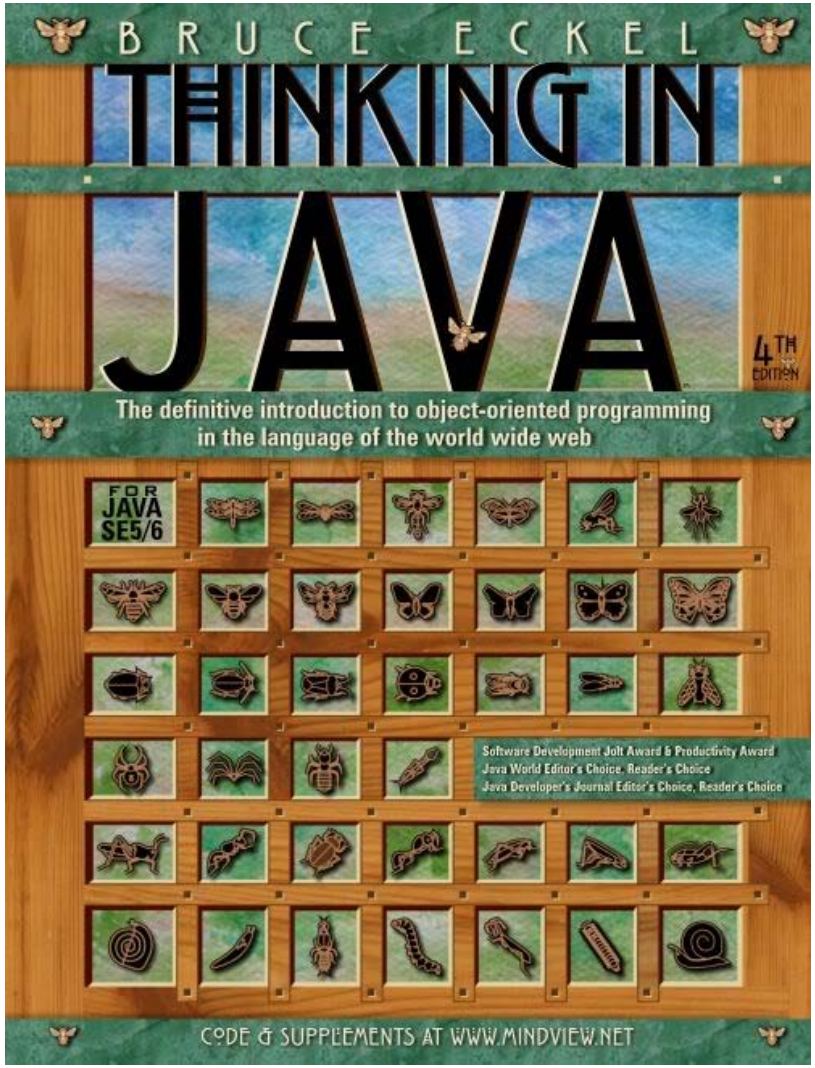
- B. Eckel. Java编程思想 (Thinking in Java), 机械工业出版社, 2016. — 全部
- J. Bloch. Effective Java 中文版, 机械工业出版社, 2009. — 全部
- GoF. 设计模式：可复用面向对象软件的基础 (Elements of Reusable Object-Oriented Software), 机械工业出版社, 2017. — 第5-6-8章
- S. McConnell. 代码大全 (Code Complete), 电子工业出版社, 2006. — 全部
- R. Martin. 代码整洁之道 (Clean Code: A Handbook of Agile Software Craftsmanship), 人民邮电出版社, 2010. — 第4-7-9章
- J. Shirazi. Java Performance Tuning, 2nd Edition, O'Reilly, 2003. — 第8章
- M. Fowler, et al. 重构：改善既有代码的设计 (Refactoring: Improving the Design of Existing Code), 人民邮电出版社, 2010. — 第9章

推荐阅读材料

- R. Pressman. 软件工程--实践者的研究方法 (Software Engineering: A Practitioner's Approach, 7th edition), 机械工业出版社, 2011. — 第1-2-7章
- P. Butcher. 软件调试修炼之道 (Debug It: Find, Repair, and Prevent Bugs in Your Code), 人民邮电出版社, 2011. — 第7章
- S. Oaks. Java性能权威指南 (Java Performance: The Definitive Guide). 中国工信出版集团, 2017. — 第8章
- P. Smith. 深入理解软件构造系统: 原理与最佳实践 (Software Build Systems: Principles and Experiences). 机械工业出版社, 2012. — 第2章
- B. Goetz. Java Concurrency in Practice, Addison-Wesley, 2006. — 第10章
- A. Oram, G. Wilson. 代码之美 (Beautiful Code), 机械工业出版社, 2009. — 课外

推荐阅读材料


计 算 机 科 学 丛 书
Java编程思想
(美) Bruce Eckel 著 陈昊鹏 译
Thinking in Java
Fourth Edition
BRUCE ECKEL
THINKING IN
JAVA
第4版
机械工业出版社
China Machine Press


BRUCE ECKEL
THINKING IN
JAVA
4TH EDITION
The definitive introduction to object-oriented programming
in the language of the world wide web
FOR JAVA SE5/6
Software Development Jolt Award & Productivity Award
Java World Editor's Choice, Reader's Choice
Java Developer's Journal Editor's Choice, Reader's Choice
CODE & SUPPLEMENTS AT WWW.MINDVIEW.NET

考核方式

- **平时成绩：5%**
  - 阅读相关书籍和论文，思考教师提出的问题，参与课堂交流讨论；
  - 针对教师提出的讨论问题，课后阅读材料，或对实验进展过程遇到的问题和经验教训进行总结思考，以文字形式发表自己的见解，以互联网上公开的博客形式发表；
  - 3月2日前，各班长提交班级所有学生个人博客表 https://cms.hit.edu.cn/mod/resource/view.php?id=23508
- **实验：35%**
  - 共6个，个人完成；
  - 现场检查、提交实验报告/实验代码至CMS/GitHub；
- **期末考试：60%**
  - 闭卷

关于课堂讨论

- 分为两类：
  - 集中式讨论：有明确的主题，教师提前给出问题，学生提前准备，形成观点，上课时由教师引导进行讨论；
  - 随堂式讨论：讲课过程中，教师根据所讲内容抛出问题，学生阐述观点。

- “翻转课堂”：课堂授课时间较少，学生需要提前阅读教师指定的教材、讲义、论文等，课堂上就其中某些问题进行研讨；

- 根据课堂上提出的讨论问题，事后形成系统化的思考，撰写个人博客。

关于实验

- 共6个实验，均为单人完成；
- 28学时实验课，课上+课后完成；
- 按照提交时间、代码/模型的质量、实验报告的质量、TA现场验收的质量进行打分；
- 6次成绩加权平均，得到总成绩。
- **Deadline: 各截止周的周日夜间23:55（CMS+GitHub）**

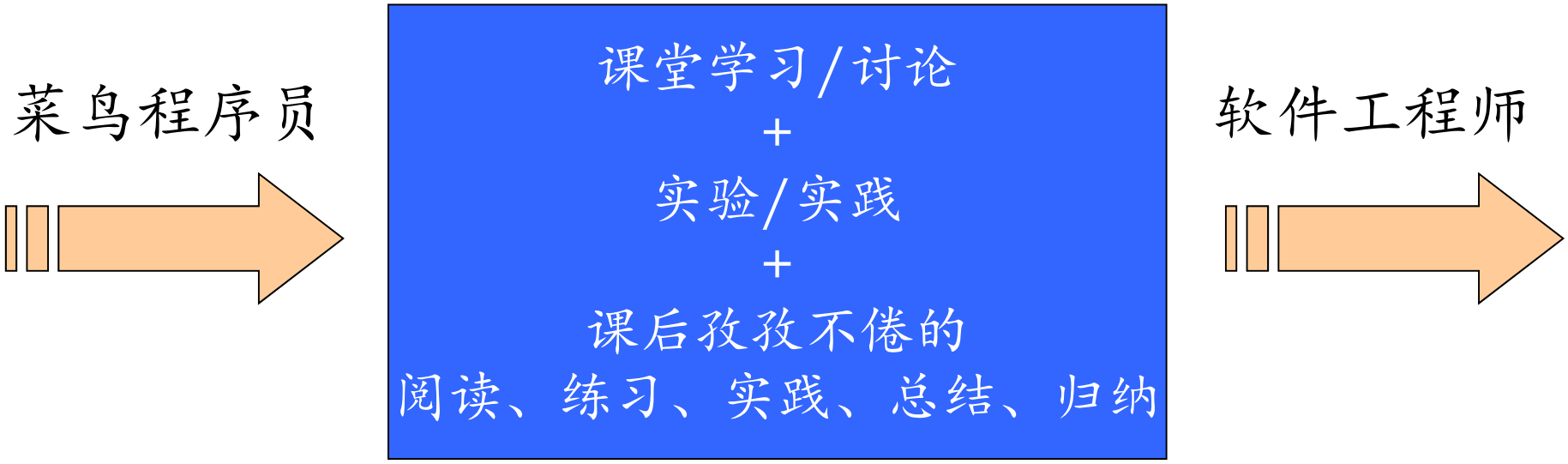
序号	实验内容	覆盖章节	实验课时间	提交截止日期	分数
1	Java编程基础/测试/构建基础	第1/2/7章	1-2周	第3周	8%
2	ADT/OOP	第3章	3-5周	第6周	20%
3	代码可维护性/扩展/复用	第5/6章	6-8周	第10周	30%
4	健壮性/调试	第7章	9-11周	第12周	15%
5	代码静态/动态评审与优化	第4/8/9章	12-13周	第14周	12%
6	多线程编程	第10章	14-15周	第16周	15%

关于实验

- 在Java+Eclipse+Git环境下进行；
- 若使用其他编程语言和编程环境，无法得到TA的有效指导，但实验要求必须全部完成，故需要自己搭建相应的开发环境与工具；
- 实验要求提前2周开放，学生可提前准备好相应的开发环境，实验课上以开发+Q&A为主；
- 代码与实验报告需在截止日期前提交至CMS/GitHub，延期不接收通过Email等其他方式的提交；
- 进行抄袭检测，若有抄袭出现，双方均无成绩；
- TA在实验课上抽查学生代码，学生口头阐述实验思路(代码、实验步骤、实验数据等)，若无法解释清楚，视为抄袭他人；
- TA课后阅读学生提交的实验报告，结合现场抽查结果、自动测试、人工评判代码，进行打分。

总结属于自己的“最佳实践”

- 多动手、多实践，方可成为合格的“程序员”；
- 实践越多、写的代码越多、参与的项目越大，积累经验越多；
- 首先遵循他人提出的“最佳实践”，进而创造自己的“最佳实践”；
- 从“菜鸟程序员”成长为“软件工程师”。



如何学习该课程

- **时刻关注课程日历，了解课程的整体进度安排，尤其是各实验的上课时间、现场检查时间、提交时间；**
  - https://cms.hit.edu.cn/mod/resource/view.php?id=23506
  - 提前搭建好实验环境，学习实验所用的工具，提前开始实验，实验课上用于与TA的交流，答疑解惑，并接受验收。
  - 单纯使用2-3学时的实验课无法完成实验。
- **提前阅读下一次课程的待讲授内容，阅读教材相关章节，进行预习；**
  - “需要我学习的知识，老师未必会在课堂上去讲”
  - 课堂上讲思想和难点，仅靠听课无法获得全部的考试点
  - 需要阅读大量的辅助教材
- **对下一次课要讲的内容，提前阅读资料做好准备。**

程序员的修炼之道

持续深入某个技术领域，坚持学习和总结

不断积累自己的代码

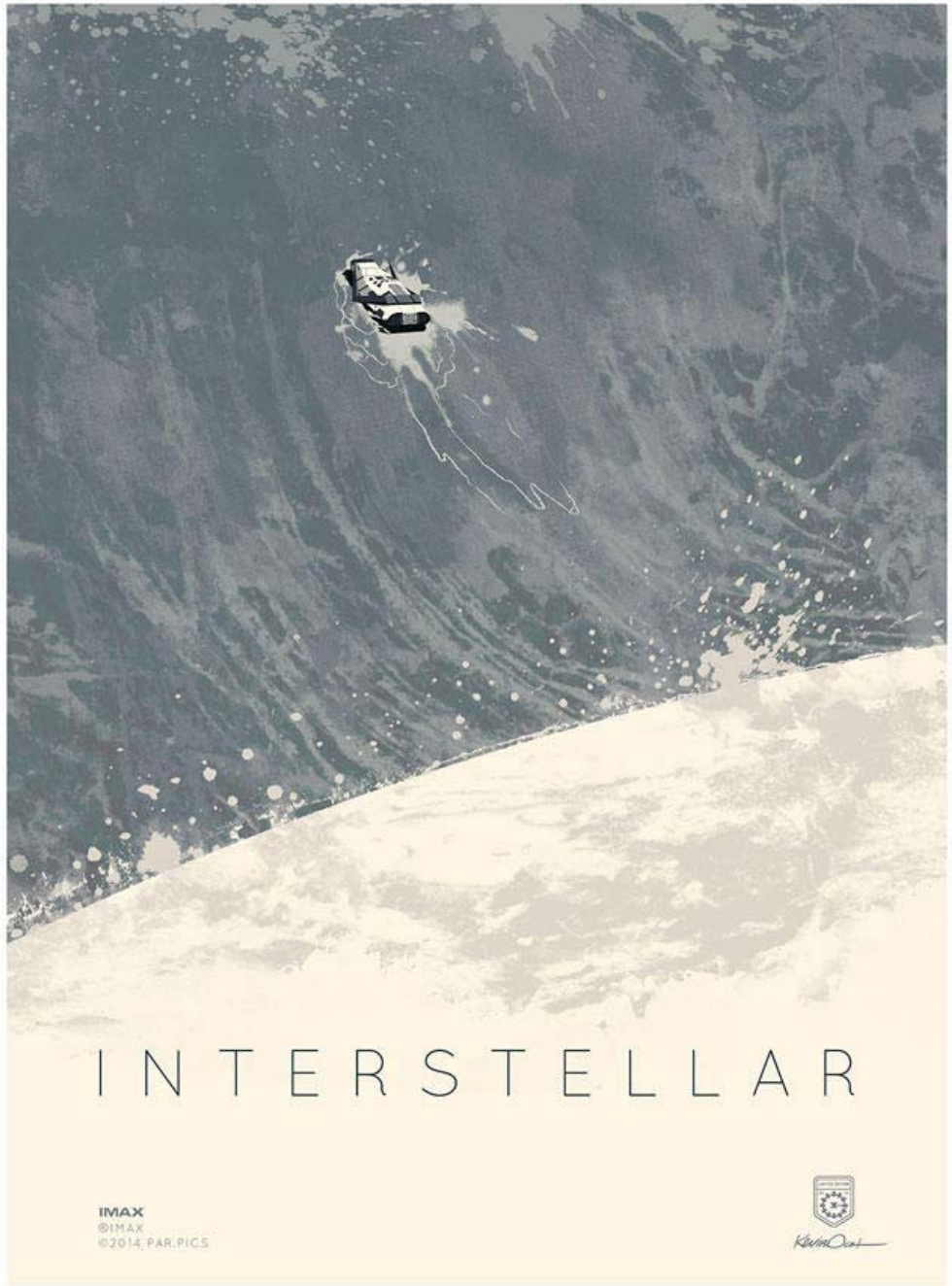
github
SOCIAL CODING

stackoverflow

与其他程序员交流经验和教训

记录你的成绩和经历

LinkedIn


INTERSTELLAR
IMAX
©IMAX
©2014 PAR.PICS

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

The end

February 26, 2018