JAVA程序设计

潘微科

教师信息

- 潘微科
- Office: 沧海校区致腾楼(计算机与软件学院)724室
- Email: panweike@szu.edu.cn
- Office hour: 周一下午14:00-16:00

- 更多信息：http://csse.szu.edu.cn/staff/panwk/

课程信息（周五下午班）

- 基本信息：
  - 课程号：150003004**[04]**
  - 课程名称：Java程序设计
  - 课程类别：学科专业选修
  - 学分：3学分（理科学分）
  - 学时：72学时（36课堂+36实验）
  - 主选班级：**2020计算机科学与技术01，2020软件工程02**
  - 课堂：每周五**第7-8节14:15-15:40**，致理楼**L1-206**教室
  - 实验：每周五**第9-10节16:00-17:25**，致腾楼**326**实验室
- QQ群：“Java程序设计【周五下午班】(2021Fall)” **732752940**
- 助教：**黄炳森**
- 先修课程：《程序设计基础》或《面向对象程序设计》
- 开发环境：JDK, Eclipse/MyEclipse/IDEA, Apache Tomcat, MySQL

群名称：Java程序设计【周五下午班】
群　号：732752940

课程信息（周五晚上班）

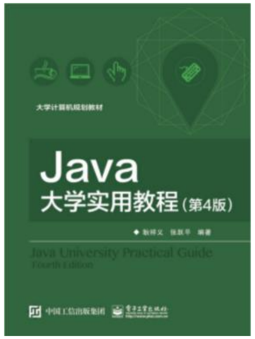
群名称：Java程序设计【周五晚上班】
群　号：734707629

- 基本信息：
  - 课程号：150003004**[02]**
  - 课程名称：Java程序设计
  - 课程类别：学科专业选修
  - 学分：3学分（理科学分）
  - 学时：72学时（36课堂+36实验）
  - 主选班级：**2020计算机科学与技术04（计算机国际班），2020计算机科学与技术05（高性能计算特色班）**
  - 课堂：每周五**第11-12节19:00-20:20**，致理楼**L2-410**教室（远程互动型智慧教室，含丽湖校区选课学生“西-204”）
  - 实验：每周五**第13-14节20:30-21:45**，致腾楼**326**实验室
- QQ群：“Java程序设计【周五下午班】(2021Fall)” **734707629**
- 助教：羊恩跃
- 先修课程：《程序设计基础》或《面向对象程序设计》
- 开发环境：JDK, Eclipse/MyEclipse/IDEA, Apache Tomcat, MySQL

教材(1/5)

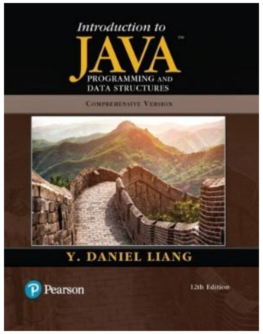
- UOOC联盟指定参考书（深圳大学教材出版基金资助）：
  - 张席. 《Java语言程序设计教程》.西安电子科技大学出版社, 2015.8.

教材(2/5)



- 教材：
  - 耿祥义.《**Java大学实用教程**》（第4版）. 电子工业出版社, 2017.3.
  - 张跃平,耿祥义,雷金娥. 《Java大学实用教程学习指导》（第3版）. 电子工业出版社, 2012.8.

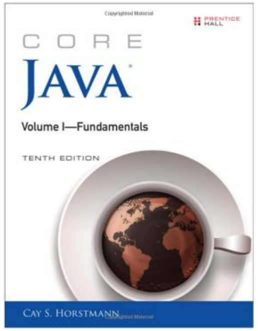
教材(3/5)

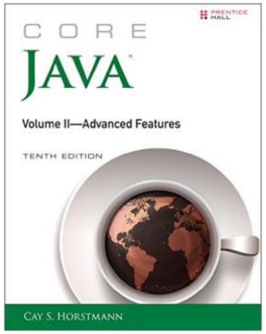


- 参考教材：
  - Y. Daniel Liang, Margaret L. Lial. **Introduction to Java Programming and Data Structures**, Comprehensive Version (**12th Edition**). Pearson, 2019.12.

  英文影印版：
  - Y. Daniel Liang. **Java语言程序设计:基础篇**（英文版 **第11版**）．机械工业出版社, 2020.6.
  - Y. Daniel Liang. **Java语言程序设计与数据结构:进阶篇**（英文版 **第11版**）．机械工业出版社, 2020.6.

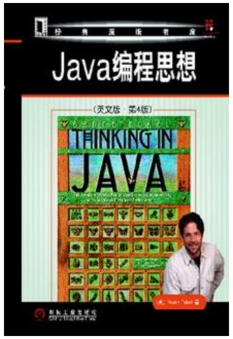
教材(4/5)





- 其他教材：
  - Cay S. Horstmann. **Core Java Volume I--Fundamentals** (10th Edition). Prentice Hall. 2016.1.（人民邮电出版社2016.6出版了英文影印版）
  - Cay S. Horstmann. **Core Java, Volume II--Advanced Features** (10th Edition). Prentice Hall. 2016.12. （人民邮电出版社2017.6出版了英文影印版）
  - 官方主页：https://horstmann.com/corejava/index.html
  - 注: 英文最新是第11版, 影印版尚未引进, 中译版已经出版.

教材(5/5)



- 其他教材：
  - Bruce Eckel. **Thinking in Java** (4th Edition). Prentice Hall, 2006.2. 注：机械工业出版社 2007.4出版了影印版, 2007.6出版了中译本（陈昊鹏 译）.

几个基本问题

- WHY：我为什么要学“Java程序设计”？
  - 在工业界和学术界的重要性不言而喻，是最受欢迎的语言之一
- WHAT：我能从“Java程序设计”课程学到什么?
  - Java编程的基础知识
- HOW：我如何能学好“Java程序设计”？
  - 教材：课堂教学
  - 实验：实际开发
  - 多查阅网上的资料，多动手（JDK, Eclipse/MyEclipse/IDEA）

一个更有挑战性的问题

- WHY：网上有那么多公开课/在线视频，我为什么还要到课堂来学？
  - 学堂在线 http://www.xuetangx.com/
  - 中国大学MOOC https://www.icourse163.org/
  - https://www.coursera.org/
  - https://www.edx.org/
  - https://www.udacity.com/

  - 如果时间允许，多渠道学习（互补性），有选择有目的（不要为了学习而学习）

线下课堂安排

- 与MOOC视频的对应关系
  - Ch1: Java语言概述 [MOOC-Ch1]
  - Ch2: 基本数据类型和数组 [MOOC-Ch2]
  - Ch3: 运算符、表达式和语句 [MOOC-Ch2]
  - Ch6: 字符串和正则表达式 [MOOC-Ch5]
  - Ch4: 类和对象 [MOOC-Ch3]
  - Ch5: 继承、接口和泛型 [MOOC-Ch4, 6, 7]
  - Ch7: 常用实用类 [MOOC-Ch6]
  - Ch8: 线程 [MOOC-Ch10]
  - Ch9: 输入流和输出流 [MOOC-Ch8]
  - Ch10: 图形用户界面设计 [MOOC-Ch9]
  - Ch11: Java中的网络编程 [MOOC-Ch11]

线下课堂安排

- 与"Introduction to Java Programming (10th Edition)"的对应关系
  - Ch1: Java语言概述 [IJP-Ch1-8]
  - Ch2: 基本数据类型和数组 [IJP-Ch1-8]
  - Ch3: 运算符、表达式和语句 [IJP-Ch1-8]
  - Ch6: 字符串和正则表达式 [IJP-Ch1-8]
  - Ch4: 类和对象 [IJP-Ch9-11, 19]
  - Ch5: 继承、接口和泛型 [IJP-Ch9-11, 19]
  - Ch7: 常用实用类 [IJP-Ch20-21]
  - Ch8: 线程 [IJP-Ch30]
  - Ch9: 输入流和输出流 [IJP-Ch12, 17]
  - Ch10: 图形用户界面设计 [IJP-Ch14-16]
  - Ch11: Java中的网络编程 [IJP-Ch31]

”Introduction to Java Programming (10th/11th/12th Edition)”

- Table of Contents (1/5)
  - Ch1 Introduction to Computers, Programs, and Java
  - Ch2 Elementary Programming
  - Ch3 Selections
  - Ch4 Mathematical Functions, Characters, and Strings
  - Ch5 Loops
  - Ch6 Methods
  - Ch7 Single-Dimensional Arrays
  - Ch8 Multidimensional Arrays

”Introduction to Java Programming (10th/11th/12th Edition)”

- Table of Contents (2/5)
  - Ch9 Objects and Classes
  - Ch10 Object-Oriented Thinking
  - Ch11 Inheritance and Polymorphism
  - Ch12 Exception Handling and Text I/O
  - Ch13 Abstract Classes and Interfaces

  - Ch14 JavaFX Basics
  - Ch15 Event-Driven Programming and Animations
  - Ch16 JavaFX UI Controls and Multimedia
  - Ch17 Binary I/O
  - Ch18 Recursion (不要求)

”Introduction to Java Programming (10th/11th/12th Edition)”

- Table of Contents (3/5)
  - Ch19 Generics
  - Ch20 Lists, Stacks, Queues, and Priority Queues
  - Ch21 Sets and Maps
  - Ch22 Developing Efficient Algorithms (不要求)
  - Ch23 Sorting (不要求)
  - Ch24 Implementing Lists, Stacks, Queues, and Priority Queues (不要求)
  - Ch25 Binary Search Trees (不要求)
  - Ch26 AVL Trees (不要求)
  - Ch27 Hashing (不要求)
  - Ch28 Graphs and Applications (不要求)
  - Ch29 Weighted Graphs and Applications (不要求)

”Introduction to Java Programming (10th/11th/12th Edition)”

- Table of Contents (4/5)

  10th Edition:

  – Ch30 Multithreading and Parallel Programming
  – Ch31 Networking
  – Ch32 Java Database Programming (不要求)
  – Ch33 JavaServer Faces (不要求)

  11th/12th Edition:

  – Ch 30 Aggregate Operations for Collection Streams (不要求)

”Introduction to Java Programming (10th Edition)”

- Table of Contents (5/5)
  - Ch34 Advanced JavaFX
  - Ch35 Advanced Database Programming
  - Ch36 Internationalization
  - Ch37 Servlets
  - Ch38 JavaServer Pages
  - Ch39 Web Services
  - Ch40 2-4 Trees and B-Trees
  - Ch41 Red-Black Trees
  - Ch42 Testing Using JUnit

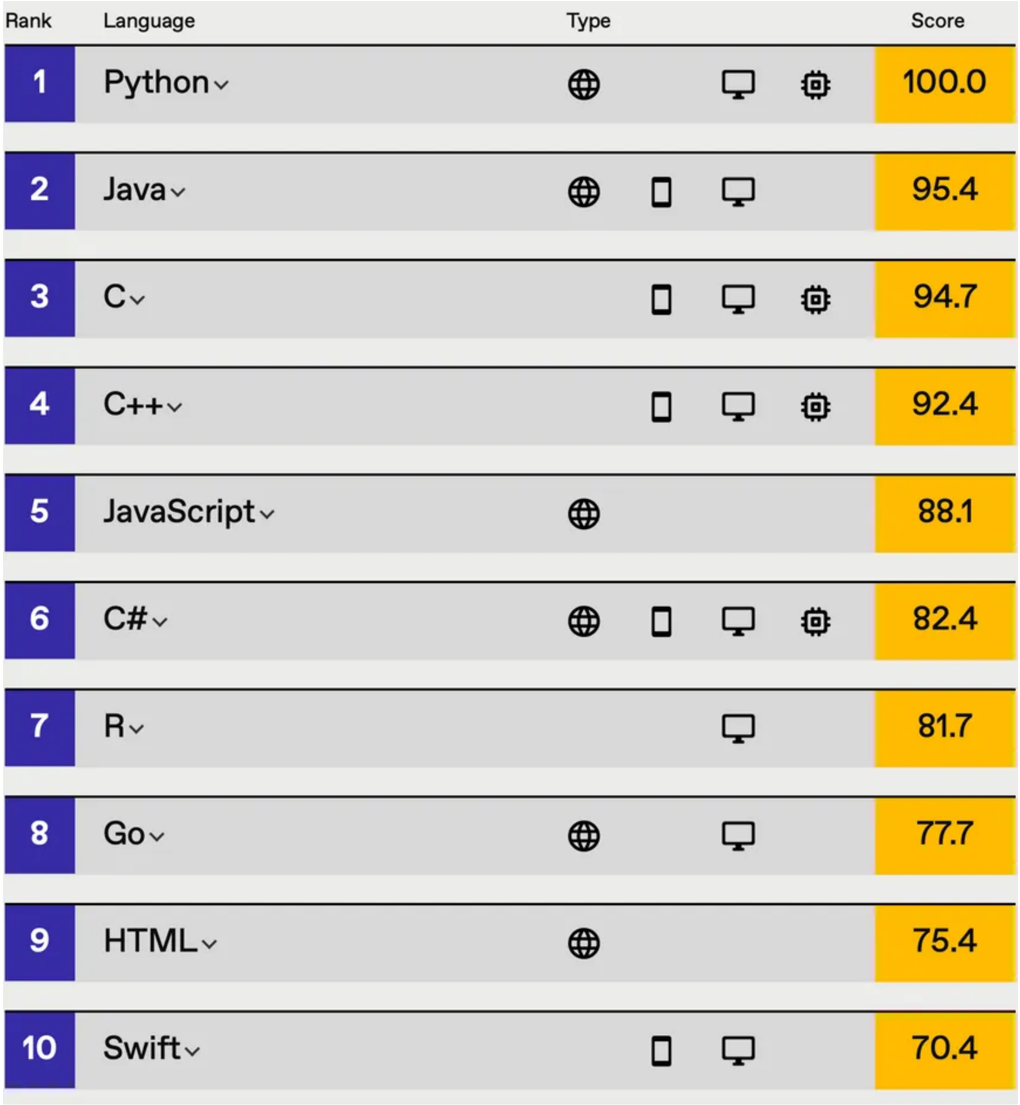
成绩评定

- 课程实验(30%)：由主讲教师自行从6次“必实验”统计
- 课程作业(10%)：由主讲教师自行从6次“选实验”统计
- 课程视频(10%)【根据MOOC网站导出的数据】
- 课程测验(5%)【根据MOOC网站导出的数据】
- 课堂讨论(5%)
- 线上期末考试(10%)【根据MOOC网站导出的数据】
- 线下期末考试(30%)

特别说明

- 感谢：教材《Java大学实用教程》的作者在华信教育资源网提供PowerPoint讲义等资料！

- 说明：本课程所使用的讲义，是在公开的讲义和学院其他老师的讲义上修改的。

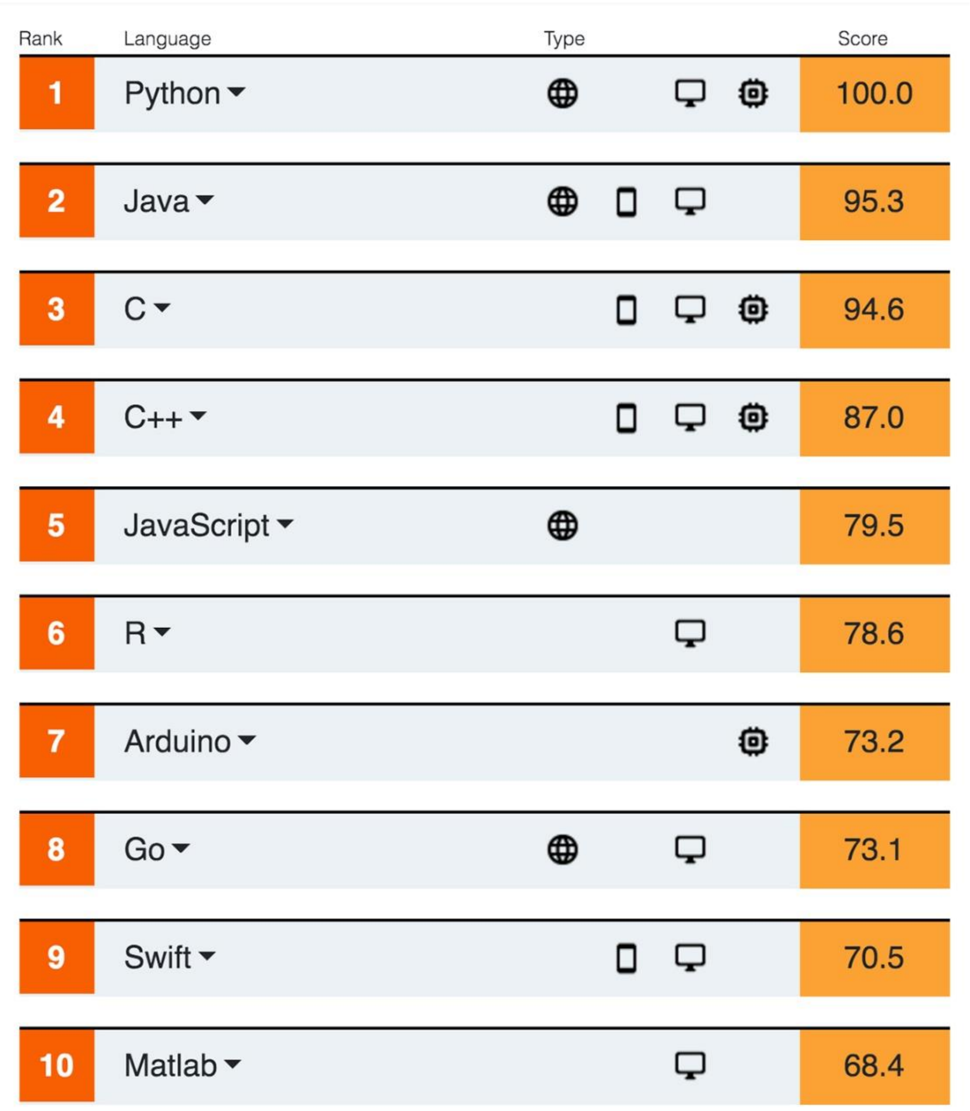


Rank	Language	Type				Score
1	Python⌄	🌐		🖥	▣	100.0
2	Java⌄	🌐	📱	🖥		95.4
3	C⌄		📱	🖥	▣	94.7
4	C++⌄		📱	🖥	▣	92.4
5	JavaScript⌄	🌐				88.1
6	C#⌄	🌐	📱	🖥	▣	82.4
7	R⌄			🖥		81.7
8	Go⌄	🌐		🖥		77.7
9	HTML⌄	🌐				75.4
10	Swift⌄		📱	🖥		70.4

No.2

Top Programming Languages 2021 by ***IEEE Spectrum***

https://spectrum.ieee.org/top-programming-languages-2021



Rank	Language	Type	Score
1	Python ▾	🌐 🖥 ▣	100.0
2	Java ▾	🌐 📱 🖥	95.3
3	C ▾	📱 🖥 ▣	94.6
4	C++ ▾	📱 🖥 ▣	87.0
5	JavaScript ▾	🌐	79.5
6	R ▾	🖥	78.6
7	Arduino ▾	▣	73.2
8	Go ▾	🌐 🖥	73.1
9	Swift ▾	📱 🖥	70.5
10	Matlab ▾	🖥	68.4

No.2

Top Programming Languages 2020 by ***IEEE Spectrum***

https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020

A Brief History of Java

- 1991: A group of Sun Microsystems engineers, led by Patrick Naughton and James Gosling
- 1995: SunWorld’95, the HotJava browser
- 1996: Java 1.0
- …
- 2004: Java 5.0
- 2006: Java 6
- 2009-2010: Sun Microsystems was acquired by Oracle
- 2011: Java 7
- 2014: Java 8
- 2017: Java 9
- 2018: Java 10, 11
- 2019: Java 12, 13
- 2020: Java 14, 15
- 2021: Java 16, 17

http://en.wikipedia.org/wiki/Java_version_history

Java As A Programming Platform

- Java is a whole **platform (平台)**:
  - a good language (e.g., pleasant syntax and comprehensive semantics)
  - a huge library (e.g., fancy graphics, networking and database access, data mining package, recommender systems, etc.)
  - a high-quality execution environment that provides services such as security, portability across operating systems, and automatic garbage collection

Core Java Volume I--Fundamentals.

Buzzwords (1/3)

- **Object oriented**: similar to that of C++.
- **Simple**: e.g., there is no pointer arithmetic (指针运算).
- **Network-savvy**: e.g., access objects across the Internet via URLs is similar to access a local file system.
- **Multithreaded**: e.g., multithreading is convenient.

Core Java Volume I--Fundamentals.

Buzzwords (2/3)

- **Robust**: Java has a pointer model that eliminates the possibility of overwriting memory and corrupting data.

- **Secure**: Java enables the construction of virus-free, tamper (篡改)-free systems, which avoids attacks.

- **Dynamic**: e.g., finding out the runtime information is straightforward.

Core Java Volume I--Fundamentals.

Buzzwords (3/3)

- **Portable**: e.g., Strings are saved in a standard Unicode format.
- **Architectural neutral**: the java compiler (编译器) generates an architecture-neutral object file format – bytecodes (字节码).
- **Interpreted**: the java interpreter (解释器) can **execute Java bytecodes** directly **on any machine** to which the interpreter has been ported.
- **High performance**: a just-in-time compiler can monitor which code is executed frequently and optimize just that code for speed.

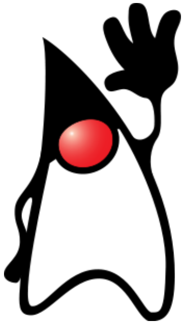
Core Java Volume I--Fundamentals.

More about Java

- http://en.wikipedia.org/wiki/Java_(programming_language)
- Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies (依赖性) as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

James Gosling,
creator of Java

Logo



Duke (公爵), Mascot (吉祥物)