
Introduction to Java Programming

Massimo Felici

Conrad Hughes

<http://www.inf.ed.ac.uk/teaching/courses/ijp>



Module aims

By the end of the course you will be able to:

- Read and write Java programs
- Integrate functionality from 3rd party libraries with your own code
- Create simple graphical interfaces
- Use appropriate software engineering techniques and tools to make your lives easier
- Tackle your summer projects with confidence!

Who should take IJP?

If you can program
but have no experience of
object-oriented programming,
IJP is for you.

If you have never programmed before

- You can take IJP, but you will find it very difficult.
- You may have to spend more time programming outside of the labs.
- You will probably pass but with a low mark.
- There have been exceptions!

Who should not take IJP?

- **If** you have completed a Java programming course at another university
- **or** you have used Java in industry
- **or** you have used Java for a large university project or dissertation
- **or** you are competent to program in another object-oriented language, for example C++
- **then** you should get an exemption from your specialism advisor.

See programming requirement in course guide:

<http://www.inf.ed.ac.uk/teaching/years/msc/courseguide10.html>

Assessment

- **Participate in the IJP PeerWise site (3%)** — contribute at least three multi-choice questions of your own creation, answer at least twenty, and provide feedback on at least twenty.
- **2 multiple-choice tests (10% each)** taken in lab sessions (we expect to include a few of the best PeerWise contributions!).
- **2 assignments (34% and 43% respectively)** — programming work, code to be submitted for assessment.
- **No exam!**

The Edinburgh Marking Scale

Score	Grade	Interpretation
≥ 70	A	Excellent
60 – 69	B	Very good
50 – 59	C	Good
40 – 49	D	Satisfactory for Diploma, but inadequate for MSc
< 40	E	Unsatisfactory

Course activities

- No Lectures!
- Instead, study the course textbook and screencasts provided by the university in your own time.
- Attend two labs per week, each two hours long, to work on programming exercises from the course textbook.
- Keep up-to-date with the PeerWise site, to see how well you're doing and help others do the same.

PeerWise

- Online repository of multi-choice questions written, answered, rated and discussed by students.
- Writing a question, coming up with good “distractors” and explaining the answer improves your understanding of the topic.
- Answering questions helps you to remember material, and to discover topics on which you need to work harder.
- Helps us discover topics which are causing the class most difficulty.
- Has been found to help with revision: initial studies indicate that its use correlates with improved grades.

Lab Times

4 Groups: IJP 1, IJP 2, IJP 3 and IJP 4

	Monday	Tuesday	Wednesday	Thursday	Friday
0900 – 0950					
1000 – 1050	IJP 1			IJP 1	
1110 – 1200	IJP 1	IJP 4		IJP 1	IJP 4
1210 – 1300		IJP 4			IJP 4
1305 – 1355					
1400 – 1450	IJP 2			IJP 2	
1500 – 1550	IJP 2			IJP 2	
1610 – 1700	IJP 3			IJP 3	
1710 – 1800	IJP 3			IJP 3	

Venue: Appleton Tower, 5.08 (Computer Lab South)

Help

- **Classmates**
- **Demonstrators**
- **Course website:**

<http://www.inf.ed.ac.uk/teaching/courses/ijp>

- **Email list:** ijp-students@inf.ed.ac.uk
- **InfBase:** Appleton Tower 5.07, at scheduled times.

Course Text

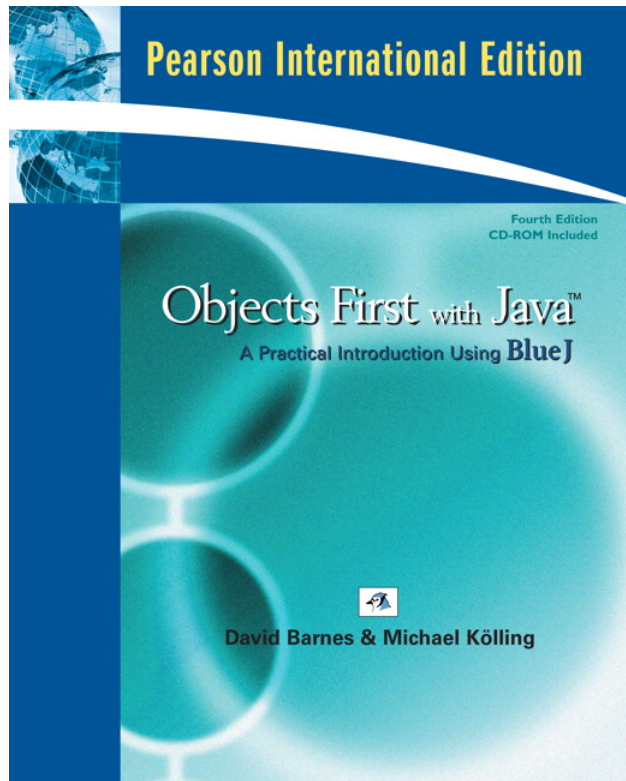
David J. Barnes & Michael Kölling

Objects First with Java:
A Practical Introduction using BlueJ

Fourth Edition
Prentice Hall / Pearson Education, 2008

ISBN-10: 0-13-700562-8 or 0-13-606086-2

ISBN-13: 978-0-13-700562-8 or 978-0-13-606086-4



You have to buy a copy.
Bring your textbook to the labs!

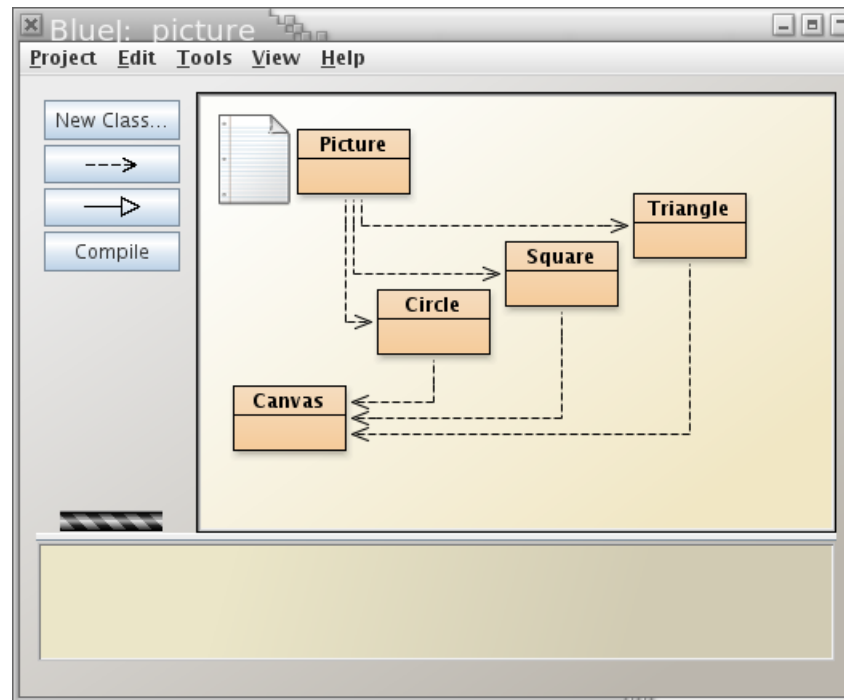
Why learn Java?

- Simple — designed from the beginning to be object-oriented
- Cross platform (write once, run anywhere)
- Commonly used
- Wide range of supporting libraries

Please read **The Java Technology Phenomenon:**

<http://download.oracle.com/javase/tutorial/getStarted/intro/index.html>

BlueJ



<http://www.bluej.org/>

Teaching approach

- **Objects first:** learn underlying concepts early to avoid getting bogged down in syntax.
- **Learn by:** executing, reading, modifying and extending example programs.
- No tediously simple “Hello, world!” programs.

Course overview (1)

- Objects and classes
- Understanding class definitions
- Object interaction
- Grouping objects
- More sophisticated behaviour — libraries
- Well-behaved objects — testing, maintaining, debugging
- Designing classes

Course overview (2)

- Inheritance
- Polymorphism
- Extendable, flexible class structures
- Building graphical user interfaces
- Handling errors
- Designing applications

Demo. . .

What next? (including Homework!)

1. Sign up to your preferred tutorial group:
<http://www.mysignup.com/ijpgroups>
2. Come to one of the PeerWise tutorials (Thu/Fri 11am, AT5.05).
3. Install BlueJ (version 3.02).
4. Download the example programs from the BlueJ book.
5. Take the BlueJ tutorial:
<http://www.bluej.org/doc/documentation.html>
Chapters 3 and 4 (later chapters are more advanced).
6. Start coming to tutorials next week!