Spring-11: PDC in CS1/2 and a mobile/cloud intermediate software design course

J. Kaylor • K. Läufer • C. Sekharan • G. K. Thiruvathukal
Dept. of Computer Science • Loyola University Chicago • USA
www.cs.luc.edu • laufer@cs.luc.edu

Abstract/Current Focus
AY 2012-13: We are moving PDC topics further down into CS1 and CS2, fleshing out PDC coverage in our intermediate object-oriented development course (CS 313), and stepping up evaluation.

Prior Outcomes
Spring 2011: We implemented three three-week PDC course modules (20% of our 15-week semester) targeting three required courses taken in the second year.
AY 2011-12: We implemented four three-week advanced PDC course modules in programming and distributed computing targeting electives typically offered every three semesters.

Institutional Profile
• Urban, private, Jesuit, liberal arts, ~16k students
• College of Arts and Sciences, ~8k students
• Dept. of Computer Science, ~250 students
• 10 FT faculty: 9 CS, 1 bioinft (1/2 FTE), 1 algebra (1/4 FTE)
• ~140 undergrad majors in CS, SE, IT, Comm. Networks & Security,
• <10 master’s students in CS, SE, IT
• External funding: NSF S-STEM, NSF BPC lead institution, NSF research grants, industry donations and grants
• Very early adopters: ’97 concurrency course, OOPSLA ’98 edx symp paper, HP/PC book

Carnegie Classification
• Level: 4-year or above
• Control: Private not-for-profit
• Undergraduate Instructional Program: Bal/HGC
• Graduate Instructional Program: CompDoc*/Med/Vet
• Enrollment Profile: MU
• Undergraduate Profile: FT4/MS/HTI
• Size and Setting: L4/R
• Basic: RUH: Research Universities (high research activity)
• Community Engagement: Curricular, Outreach, Partnerships
*CS/SE/IT: up to masters’ level

Intermediate Object-Oriented Development
• Loyola course number: COMP 313
• since fall 2011: C++, emphasis on PDC topics
• since fall 2012: Java with Android as highly effective context for studying concurrency and distributed computing topics (C and A levels)
• double 18-hour PDC modules covers
• concurrence and coordination
• design forces: safety, liveness, performance
• external events and internal timers
• background threads
• progress reporting and cancellation
• offloading computation from mobile device to cloud
• throughout-latency tradeoff
• example to illustrate these topics (see screenshot)
• Android app: brute-force prime checker
• remote task returns quickly, local one is still churning

Future Plans
• Computer Systems/CS3
• Loyola course number: COMP 264
• offered every spring
• envisioning PDC module with suitable architecture, programming, and cross-cutting topics

Algorithms/CS7
• Loyola course number: COMP 363
• offered every fall
• developing PDC module: models of computation and complexity, basic algorithmic paradigms, and specific problems and their algorithmic solutions
• Evaluation
• Once our course modules have stabilized, we will need to measure their effectiveness longitudinally over a three- to five-year period.
• Refine our current evaluation approach by working with Loyola’s Center for Science & Math Education, as well as the TCPF and fellow adopters.
• Dissemination: workshops for subsequent adopters in the Midwest