

SWEN423 Object-Oriented Paradigms

Reading list for SWEN423 Object-Oriented Paradigms

View Online



Aldrich, J. (2013). The power of interoperability. Proceedings of the 2013 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software - Onward! '13, 101-116. <https://doi.org/10.1145/2509578.2514738>

Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77. <https://doi.org/10.1109/2.796139>

Beck, K. (2001). Manifesto for Agile Software Development. <http://agilemanifesto.org/>

Beck, K., & Cunningham, W. (1989). A laboratory for teaching object oriented thinking. Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications - OOPSLA '89, 1-6. <https://doi.org/10.1145/74877.74879>

Bolz, C. F., Diekmann, L., & Tratt, L. (2013). Storage strategies for collections in dynamically typed languages. Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications - OOPSLA '13, 167-182. <https://doi.org/10.1145/2509136.2509531>

Bracha, G., & Ungar, D. (2004). Mirrors. Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications - OOPSLA '04. <https://doi.org/10.1145/1028976.1029004>

Bracha, G., von der Ahé, P., Bykov, V., Kashai, Y., Maddox, W., & Miranda, E. (2010). Modules as Objects in Newspeak. In T. D'Hondt (Ed.), *ECOOP 2010 – Object-Oriented Programming* (Vol. 6183, pp. 405-428). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14107-2_20

Byte Magazine - Smalltalk. (1981). 06(08). <https://archive.org/details/byte-magazine-1981-08>

Cargill, T. A. (1991). Controversy: The Case Against Multiple Inheritance in C++. *Computing Systems: A Quarterly Publication of the USENIX Association*, 4(1), 69-82. https://www.usenix.org/legacy/publications/compsystems/1991/win_cargill.pdf

Cook, W. R. (2009). On understanding data abstraction, revisited. *ACM SIGPLAN Notices*, 44(10). <https://doi.org/10.1145/1639949.1640133>

Cunningham, W. (1994). A CRC Description of HotDraw. <http://c2.com/doc/crc/draw.html>

Deep Java. (n.d.). <http://homepages.ecs.vuw.ac.nz/~tk/publications/papers/deep-java.pdf>
Ducasse, S., Nierstrasz, O., Schärli, N., Wuyts, R., & Black, A. P. (2006). Traits. *ACM*

Transactions on Programming Languages and Systems, 28(2), 331–388.
<https://doi.org/10.1145/1119479.1119483>

Ernst, E. (n.d.). The expression problem, scandinavian style. MASPEGHI 2004 - Mechanisms for Specialization, Generalization and Inheritance, 2004.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.5386&rep=rep1&type=pdf>

Feitelson, D. G., Frachtenberg, E., & Beck, K. L. (2013). Development and Deployment at Facebook. *IEEE Internet Computing*, 17(4), 8–17. <https://doi.org/10.1109/MIC.2013.25>

Foote, B., & Yoder, J. (2000). Chapter 29: Big Ball of Mud. In N. Harrison, B. Foote, & H. Rohnert (Eds.), *Pattern Languages of Program Design 4* (pp. 653–692). Addison-Wesley.
<http://www.laputan.org/mud/>

Goldberg, A., & Robson, D. (1983). *Smalltalk-80: The Language and Its Implementation*. Addison-Wesley. <http://stephane.ducasse.free.fr/FreeBooks/BlueBook/Bluebook.pdf>

Hopkins, T., & Horan, B. (1995). Chapter 26: Classes and Metaclasses. In *Smalltalk: An Introduction to Application Development Using Visualworks* (pp. 279–286). Prentice-Hall.
<http://trevor-hopkins.com/downloads/HopkinsHoran.pdf>

Johnson, R. E. (1992). Documenting frameworks using patterns. *Conference Proceedings on Object-Oriented Programming Systems, Languages, and Applications - OOPSLA '92*, 63–76. <https://doi.org/10.1145/141936.141943>

Johnson, R. E. (1997). Frameworks = (components + patterns). *Communications of the ACM*, 40(10), 39–42. <https://doi.org/10.1145/262793.262799>

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., & Griswold, W. (2001a). Getting started with ASPECTJ. *Communications of the ACM*, 44(10), 59–65.
<https://doi.org/10.1145/383845.383858>

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., & Griswold, W. G. (2001b). An Overview of AspectJ. In J. L. Knudsen (Ed.), *ECOOP 2001 — Object-Oriented Programming* (Vol. 2072, pp. 327–354). Springer Berlin Heidelberg.
https://doi.org/10.1007/3-540-45337-7_18

Lagorio, G., & Servetto, M. (2011). Strong exception-safety for checked and unchecked exceptions. *The Journal of Object Technology*, 10. <https://doi.org/10.5381/jot.2011.10.1.a1>

LaLonde, W., & Pugh, J. (n.d.). Subclassing (not-equal-to) subtyping (not-equal-to) is-a. *Journal of Object-Oriented Programming (JOOP)*, 3(5), 57–62.

Madsen, O. L., Moller-Pedersen, B., & Nygaard, K. (1993). *Object-Oriented Programming in the Beta Programming Language* (2nd edition). Pearson Education.

Marco Servetto, Alex, Potanin, David, Pearce, & Lindsay, Groves. (n.d.). Balloon Types for Safe Parallelisation over Arbitrary Object Graphs.
<https://www.semanticscholar.org/paper/Balloon-Types-for-Safe-Parallelisation-over-Object-Servetto-Pearce/4522464d6ef46d396737e81958fcf4129978160d?tab=abstract>

- Miller, M. S. (2006). Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control [Johns Hopkins University].
<http://erights.org/talks/thesis/index.html>
- Muschevici, R., Potanin, A., Tempero, E., & Noble, J. (2008). Multiple dispatch in practice. Proceedings of the 23rd ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications - OOPSLA '08.
<https://doi.org/10.1145/1449764.1449808>
- Paola, Giannini, Elena, Zucca, Marco, Servetto, & James, Cone. (2018a). Flexible recovery of uniqueness and immutability.
<https://www.sciencedirect.com/science/article/pii/S0304397518305668>
- Paola, Giannini, Elena, Zucca, Marco, Servetto, & James, Cone. (2018b). Flexible recovery of uniqueness and immutability.
<https://www.sciencedirect.com/science/article/pii/S0304397518305668>
- Richards, G., Hammer, C., Burg, B., & Vitek, J. (2011). The Eval That Men Do. In M. Mezini (Ed.), ECOOP 2011 – Object-Oriented Programming (Vol. 6813, pp. 52–78). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22655-7_4
- Roberts, D., & Johnson, R. (1998). Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks. Pattern Languages of Program Design 3, 471–486. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.8767>
- Safely Composable Type-Specific Languages. (n.d.).
<http://www.cs.cmu.edu/~aldrich/papers/ecoop14-tsls.pdf>
- Separating Use and Reuse to Improve Both - Details. (n.d.).
<https://programming-journal.org/2019/3/12/>
- Servetto, M., Mackay, J., Potanin, A., & Noble, J. (2013). The Billion-Dollar Fix. ECOOP 2013 – Object-Oriented Programming, 7920, 205–229.
https://doi.org/10.1007/978-3-642-39038-8_9
- Servetto, M., & Zucca, E. (2014). A meta-circular language for active libraries. Science of Computer Programming, 95, 219–253. <https://doi.org/10.1016/j.scico.2014.05.003>
- Stroustrup, B. (n.d.). Sixteen Ways to Stack a Cat. The C++ Report: The International Authority on C++ Development. http://www.stroustrup.com/stack_cat.pdf
- Stroustrup, B. (1995). Why C++ is not just an object-oriented programming language. Addendum to the Proceedings of the 10th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (Addendum) - OOPSLA '95, 1–13.
<https://doi.org/10.1145/260094.260207>
- Taivalsaari, A. (1996). On the notion of inheritance. ACM Computing Surveys, 28(3), 438–479. <https://doi.org/10.1145/243439.243441>
- The Untold Story of NotPetya, the Most Devastating Cyberattack in History | WIRED. (n.d.-a).
<https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>

- The Untold Story of NotPetya, the Most Devastating Cyberattack in History | WIRED. (n.d.-b). <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>
- Ungar, D., & Smith, R. B. (1987). Self: The power of simplicity. Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications - OOPSLA '87, 227–242. <https://doi.org/10.1145/38765.38828>
- Waldo, J. (1991). Controversy: The Case For Multiple Inheritance in C++. Computing Systems: A Quarterly Publication of the USENIX Association, 4(2), 157–171. https://www.usenix.org/legacy/publications/compsystems/1991/spr_waldo.pdf
- Wang, Y., & Oliveira, B. C. d. S. (2016). The expression problem, trivially! Proceedings of the 15th International Conference on Modularity - MODULARITY 2016, 37–41. <https://doi.org/10.1145/2889443.2889448>
- Wang, Y., Zhang, H., Oliveira, B. C. d. S., & Servetto, M. (2016). Classless Java. ACM SIGPLAN Notices, 52(3), 14–24. <https://doi.org/10.1145/3093335.2993238>
- Wyvern: A Language for Usable Design-Driven Assurance. (n.d.). <https://github.com/wyvernlang/wyvern/wiki/Wyvern%3A-A-Language-for-Usable-Design-Driven-Assurance>
- Wyvern: Impacting Software Security via Programming Language Design. (n.d.). <http://www.cs.cmu.edu/~aldrich/papers/plateau14-wyvern.pdf>
- Zenger, M., & Odersky, M. (n.d.). FOOL 2005 - Independently Extensible Solutions to the Expression Problem. <http://homepages.inf.ed.ac.uk/wadler/fool/program/10.html>