Bibliography

References

1. Sommerville, I. (2011). Introduction. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P1-26.
2. Citizens Advice. (2015). *The benefit cap - what you need to know.* Available: https://www.citizensadvice.org.uk/benefits/the-benefit-cap/the-benefit-cap-what-you-need-to-know/. Last accessed 20th Jun 2015.
3. Enfield Council. (2013). *Enfield's Homelessness Strategy 2013-2018.* Available: http://www.enfield.gov.uk/download/downloads/id/8004/enfields\_homelessness\_strategy\_2013-2018. Last accessed 20th Jun 2015.
4. Hunt, B. (2015). *UH Ethics Approval.* Available: http://www.studynet2.herts.ac.uk/ptl/common/ethics.nsf/Homepage?ReadForm. Last accessed 19th Jul 2015.
5. Cockburn, A and Highsmith. (2001). Agile Software Development: The People Factor. *Computer*. 34 (1), p131-133.
6. Huo, M, Verner, J, Zhu, L and Babar. (2004). Software quality and agile methods. *Computer Software and Applications Conference*. 1 (1), p520 - 525.
7. H, Gomaa. (2001). ICSE 01 Proceedings of the 23rd International Conference on Software Engineering. *Designing Concurrent, Distributed, and Real-Time Applications with UML*. 1 (1), p737 - 738.
8. Sommerville, I. (2011). Software processes. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P27-55.
9. A.D, Birrell and B.J, Nelson. (1984). ACM Transactions on Computer Systems. *Implementing remote procedure calls*. 2 (1), p39 - 59.
10. J.P, Martin-Flatin. (1999). Push vs. pull in Web-based network management. *Integrated Network Management, 1999. Distributed Management for the Networked Millennium. Proceedings of the Sixth IFIP/IEEE International Symposium on*. 1 (1), p3 - 18.
11. D, Habam and K.G, Shin. (2002). Application of real-time monitoring to scheduling tasks with random execution times. *Software Engineering, IEEE Transactions on*. 16 (1), p1374 - 1389.
12. M.L, Horowitz. (1991). An Introduction to Object-Oriented Database and Database Systems. *Object-Oriented Database and Database Systems*. 1 (1), p1 - 86.
13. C, Amza, A, Chanda, S, Elnijety, R, Gil, K, Rajamani, W, Zwaenepoel, E, Cecchet and J, Marguerite. (2002). Specification and implementation of dynamic Web site benchmarks. *Workload Characterization, 2002. WWC-5. 2002 IEEE International Workshop on*. 1 (1), p3 - 13.
14. Y, Cheon and G.T, Leavens. (2002). A Simple and Practical Approach to Unit Testing: The JML and JUnit Way. *ECOOP 2002 — Object-Oriented Programming*. 2374 (1), p231 - 255.
15. L, Briand and Y, Labiche. (2002). A UML-Based approach to System Testing. *Software and Systems Modeling*. 1 (1), p10 - 42.
16. Wikipedia. (2012). *Use Case Diagram.* Available: https://en.wikipedia.org/wiki/Use\_Case\_Diagram. Last accessed 10th Oct 2015.
17. Wikipedia. (2004). *Data flow diagram.* Available: https://en.wikipedia.org/wiki/Data\_flow\_diagram. Last accessed 10th Oct 2015.
18. Wikipedia. (2005). *Class Diagram.* Available: https://en.wikipedia.org/wiki/Class\_diagram. Last accessed 16th Aug 2015.
19. Wikipedia. (2004). *Entity Relationship Model.* Available: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship\_model. Last accessed 23rd Oct 2015.
20. Wikipedia. (2004). *Class-responsibility-collaboration card.* Available: https://en.wikipedia.org/wiki/Class-responsibility-collaboration\_card. Last accessed 11th Aug 2015.
21. Wikipedia. (2005). *Sequence Diagram.* Available: https://en.wikipedia.org/wiki/Sequence\_diagram. Last accessed 10th Oct 2015.
22. Wikipedia. (2002). *Storyboard.* Available: https://en.wikipedia.org/wiki/Storyboard. Last accessed 10th Oct 2015.
23. Wikipedia. (2005). *Observer pattern.* Available: https://en.wikipedia.org/wiki/Observer\_pattern. Last accessed 30th Oct 2015.
24. Wikipedia. (2002). *Singleton pattern.* Available: https://en.wikipedia.org/wiki/Singleton\_pattern. Last accessed 9th Oct 2015.
25. Wikipedia. (2003). *Strategy pattern.* Available: https://en.wikipedia.org/wiki/Strategy\_pattern. Last accessed 10th Aug 2015.
26. Wikipedia. (2004). *Creational pattern.* Available: https://en.wikipedia.org/wiki/Creational\_pattern. Last accessed 10th Aug 2015.
27. Wikipedia. (2003). *Iterator pattern.* Available: https://en.wikipedia.org/wiki/Iterator\_pattern. Last accessed 10th Aug 2015.
28. Wikipedia. (2003). *Composite pattern.* Available: https://en.wikipedia.org/wiki/Composite\_pattern. Last accessed 10th Aug 2015.
29. Wikipedia. (2001). *Inheritance (object-oriented programming.* Available: https://en.wikipedia.org/wiki/Inheritance\_(object-oriented\_programming). Last accessed 10th Aug 2015.
30. Wikipedia. (2005). *Object composition.* Available: https://en.wikipedia.org/wiki/Object\_composition. Last accessed 10th Aug 2015.
31. Wikipedia. (2002). *Object-relational mapping.* Available: https://en.wikipedia.org/wiki/Object-relational\_mapping. Last accessed 31st Aug 2015.
32. The Apache Software Foundation. (2006). *Apache JackRabbit.* Available: https://jackrabbit.apache.org/jcr/index.html. Last accessed 11th Aug 2015.
33. JBossDeveloper. (2010). *Modeshape.* Available: http://modeshape.jboss.org/. Last accessed 11th Aug 2015.
34. Wikipedia. (2002). *Microsoft Access.* Available: https://en.wikipedia.org/wiki/Microsoft\_Access. Last accessed 11th Aug 2015.
35. Wikipedia. (2001). *MySQL.* Available: https://en.wikipedia.org/wiki/MySQL. Last accessed 31st Aug 2015.
36. Wikipedia. (2001). *Apache HTTP Server.* Available: https://en.wikipedia.org/wiki/Apache\_HTTP\_Server. Last accessed 11th Aug 2015.
37. Wikipedia. (2007). *Nginx.* Available: https://en.wikipedia.org/wiki/Nginx. Last accessed 11th Aug 2015.
38. Wikipedia. (2010). *Cherokee (web server).* Available: https://en.wikipedia.org/wiki/Cherokee\_(web\_server). Last accessed 11th Aug 2015.
39. Wikipedia. (2002). *Abstract Window Toolkit.* Available: https://en.wikipedia.org/wiki/Abstract\_Window\_Toolkit. Last accessed 11th Aug 2015.
40. Wikipedia. (2003). *Swing (Java).* Available: https://en.wikipedia.org/wiki/Swing\_(Java). Last accessed 11th Aug 2015.
41. D, Matuszek. (2009). *Enhanced for Loop.* Available: https://www.cis.upenn.edu/~matuszek/General/JavaSyntax/enhanced-for-loops.html. Last accessed 27th Nov 2015.
42. Wikipedia. (2001). *JUnit.* Available: https://en.wikipedia.org/wiki/JUnit. Last accessed 10th Dec 2015.
43. Wikipedia. (2005). *Java Modeling Language.* Available: https://en.wikipedia.org/wiki/Java\_Modeling\_Language. Last accessed 10th Dec 2015.

**Further Reading**

1. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Charecterization of Distributed Systems. In:Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P17-52.
2. Sommerville, I. (2011). Distributed software engineering. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P479-507.
3. Cockburn, A. and Highsmith, J. and Bohem, B. (2001). Agile Software Development: The Business of Innovation. *Computer*. 1 (1), p131-133
4. Sommerville, I. (2011). Requirements engineering. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P82-117.
5. Sommerville, I. (2011). System modeling. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P118-146.
6. Connolly, T. and Begg, C. (2005). Normalization. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P387-414.
7. Connolly, T. and Begg, C. (2005). Entity-Relationship Modeling. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P387-414.
8. Sommerville, I. (2011). Design and implementation. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P176-204.
9. Reges, S. and Stepp, M. (2011). Graphical User Interface. In: Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Building Java Programs A Back to Basics Approach.* 2nd ed. Boston: Pearson. P846-909.
10. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Remote Invocation. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P201-246.
11. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Distributed Objects and Components. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P351-396.
12. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Transactions and Concurrency Control. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P691-742.
13. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Distributed Transactions. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P743-780.
14. Connolly, T. and Begg, C. (2005). SQL: Data Manipulation. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P112-156.
15. Connolly, T. and Begg, C. (2005). Security. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P541-571.
16. Connolly, T. and Begg, C. (2005). Transaction Management. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P572-629.
17. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Security. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P479-536.
18. Sommerville, I. (2011). Software testing. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P205-233.
19. Sommerville, I. (2011). Project management. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P593-617.
20. Sommerville, I. (2011). Project planning. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P618-650.
21. Waldo, J. (1998). Remote procedure calls and Java Remote Method Invocation. *Concurrency, IEEE*. 6 (3), P5-7.
22. Guan, H. and Ip, H. and Zhang, Y. (1998). Java-based approaches for accessing databases on the Internet and a JDBC-ODBC implementation. *Computing & Control Engineering Journal*. 9 (2), P71-78.
23. Wikipedia. (2004). *Data Flow Diagram.* Available: 17. https://en.wikipedia.org/wiki/Data\_flow\_diagram. Last accessed 16th Aug 2015.