Mindluster Platform

About the course Software Engineering SE

Software engineering is important because specific software is needed in almost every industry, in every business, and for every function. It becomes more important as time goes on â€“ if something breaks within your application portfolio, a quick, efficient, and effective fix needs to happen as soon as possible.

Computer Softwares Category's Courses

Course Lesson(99)

Lesson 1 : **what is software engineering**

Lesson 2 : **software crisis | software engineering |**

Lesson 3 : **software life cycle model | software engineering |**

Lesson 4 : **waterfall model | software engineering |**

Lesson 5 : **prototype model in software engineering**

Lesson 6 : **spiral model | software engineering |**

Lesson 7 : **changing nature of software | software engineering |**

Lesson 8 : **software quality | software engineering |**

Lesson 9 : **SEI CMM | software engineering |**

Lesson 10 : **what is iso 9001 | software engineering |**

Lesson 11 : **process flow | linear iterative | software engineering**

Lesson 12 : **Generic process model | software engineering |**

Lesson 13 : **process assessment | software engineering |**

Lesson 14 : **personal software process | software engineering |**

Lesson 15 : **team software process | software engineering |**

Lesson 16 : **Unified process | 5 Stage | Software Engineering**

Lesson 17 : **incremental model | software engineering |**

Lesson 18 : **RAD model | software engineering |**

Lesson 19 : **specialized process model | software engineering |**

Lesson 20 : **Line of Code LOC | software engineering |**

Lesson 21 : **functional point analysis | part 1 2 | SE |**

Lesson 22 : **functional point analysis | part 2 2 | SE | bhanu priya**

Lesson 23 : **information flow metric | software Engineering |**

Lesson 24 : **software metrics | software engineering |**

Lesson 25 : **software myths | software engineering |**

Lesson 26 : **software project planning | software Engineering |**

Lesson 27 : **cost estimation | software Engineering |**

Lesson 28 : **static single variate model | software engineering |**

Lesson 29 : **static multivariate model | software engineering**

Lesson 30 : **static single multivariate model | examples | Software Engineering**

Lesson 31 : **COCOMO model | part 1 3 | Software Engineering |**

Lesson 32 : **COCOMO model | part 2 3 | Software Engineering |**

Lesson 33 : **COCOMO model | part 3 3 | Software Engineering |**

Lesson 34 : **ground work establishment | part 1 2 |**

Lesson 35 : **ground work establishment | part 2 2 |**

Lesson 36 : **scenario based modeling | Software engineering |**

Lesson 37 : **risk management | Software engineering |**

Lesson 38 : **risks in project management | Software engineering |**

Lesson 39 : **requirement engineering | Software engineering |**

Lesson 40 : **Problem solving strategies | 7 steps | Requirement engineering | Software engineering |**

Lesson 41 : **data flow diagram | software engineering |**

Lesson 42 : **data dictionary | software engineering |**

Lesson 43 : **entity relationship diagram | software engineering |**

Lesson 44 : **uml model | software engineering |**

Lesson 45 : **software requirement specification | software engineering |**

Lesson 46 : **behavioral and non behavioral requirements |**

Lesson 47 : **software prototyping | software engineering |**

Lesson 48 : **prototyping in software process | SE |**

Lesson 49 : **software design | introduction |software engineering |**

Lesson 50 : **cohesion | software engineering |**

Lesson 51 : **coupling | software engineering |**

Lesson 52 : **classification of cohesion | software engineering |**

Lesson 53 : **coupling classification | Software Engineering |**

Lesson 54 : **software architecture | software engineering |**

Lesson 55 : **function oriented design | software engineering |**

Lesson 56 : **object oriented design | software engineering |**

Lesson 57 : **user interface design | software engineering |**

Lesson 58 : **architecture geners | software engineering |**

Lesson 59 : **architecturel style | introduction | software engineering |**

Lesson 60 : **architectural styles | part 1 2 | software engineering**

Lesson 61 : **architectural styles | part 2 2 | software engineering |**

Lesson 62 : **architectural design | software engineering |**

Lesson 63 : **assessing alternative architectural design | part 1 3 |**

Lesson 64 : **assessing alternative architectural design | part 2 3 |**

Lesson 65 : **assessing alternative architectural design | part 3 3 |**

Lesson 66 : **user interface golden rules | software Engineering |**

Lesson 67 : **User interface analysis Design | Software engineering**

Lesson 68 : **User interface design Models | software engineering |**

Lesson 69 : **User interface analysis| software engineering |**

Lesson 70 : **interface design steps | software engineering |**

Lesson 71 : **web design goals | Software Engineering |**

Lesson 72 : **web application design quality | Software Engineering |**

Lesson 73 : **quality dimensions for end users | SE |**

Lesson 74 : **design pyramid for webapps | software engineering |**

Lesson 75 : **component level design | OOHDM | software engineering |**

Lesson 76 : **interface design principle| part 1 3 | Software Engineering |**

Lesson 77 : **interface design principle| part 2 3 | Software Engineering |**

Lesson 78 : **interface design principle| part 3 3 | Software Engineering |**

Lesson 79 : **Aesthetic design | Software Engineering |**

Lesson 80 : **content design | software engineering |**

Lesson 81 : **navigation design | software engineering |**

Lesson 82 : **Content web app architecture |architecture design |Software Engineering**

Lesson 83 : **maintenance process | software engineering |**

Lesson 84 : **software maintenance | software engineering |**

Lesson 85 : **software maintenance models | part 1 2|**

Lesson 86 : **software maintenance model | part 2 2|**

Lesson 87 : **re engineering | defination | software engineering |**

Lesson 88 :
Cost factors Process activities configuration management | Re engineering |software engineering |

Lesson 89 : **reverse engineering | software engineering |**

Lesson 90 : **unit testing | software engineering |**

Lesson 91 : **system testing | software engineering |**

Lesson 92 : **regression testing | software engineering |**

Lesson 93 : **software testing | software engineering |**

Lesson 94 : **black box testing | software engineering |**

Lesson 95 : **white box testing | software engineering |**

Lesson 96 : **gray box testing | software engineering |**

Lesson 97 : **integration testing | part 1 3 |software engineering**

Lesson 98 : **integration testing | part 2 3 |software engineering**

Lesson 99 : **integration testing | part 3 3 |software engineering**

Related courses

Excel Power Query

Excel VBA

Excel Dashboard

Excel Course Basic to Advanced

Excel Functions

Excel Charting

for Business Contact
business@mindluster.com