

33. Software Construction Fundamentals: Standards in design.
34. Standard vs Enterprise Software Development: what's the differences?
35. Which Modular software development principles do you know?
36. Minimizing Complexity: Modularity, Prototyping, Simple Algorithms, Readability
37. Minimizing Complexity: Programming Standards.
38. Modular software development principle and Module properties.
39. Prototype development stages and Prototyping methodologies (Fast, Evolutionary).
40. What do you know about the process of Layout?
41. Waiting for changes: refactoring and reengineering.
42. What is refactoring? Which main Reasons for the need for refactoring do you know?
43. Testing: Module and Unit-testing.
44. Types of Software Testing.
45. Unit testing: main tasks.
46. Module testing: examples, benefits and challenges.
47. Main directions of Software Design.
48. Minimizing the complexity of Software Development.
49. Software development methods: main tasks.
50. Software life cycle paradigms and models. Cascading model, its benefits and challenges.
51. Software life cycle paradigms and models. Waterfall model, its benefits and challenges.
52. Software life cycle paradigms and models. Incremental model, its benefits and challenges.
53. Software life cycle paradigms and models. Evolutionary (spiral) model, its benefits and challenges.
54. Software life cycle paradigms and models. V-model, its benefits and challenges.
55. What do you know about Software engineering management?
56. The main measurements in design. Direct and Indirect measurements.
57. How to Reduce Software Development Complexity?
58. Analysis of minimizing complexity in software construction.
59. How to Simplify Code Complexity?
60. Software Development Life Cycle: The Design Phase.
61. Verifying and validating the analysis models: Functional and Structural Models.
62. Verifying and validating the analysis models: Structural and Behavioral Models.
63. Factoring Process: Abstraction and Refinement, Partitions and Collaborations.
64. Design strategies: Custom Development.
65. Design strategies: Packaged Software.
66. Design strategies: Outsourcing.
67. Different types of software testing and usage them.
68. Different types of software testing: unit tests.
69. Different types of software testing: integration tests.
70. Different types of software testing: system tests.
71. Developing documentation: system documentation and user documentation.
72. The main concepts of Object-oriented System development.
73. Basic characteristics of Object-Oriented Systems Analysis & Design.
74. Basic characteristics of object-oriented systems: methods, messages
75. Basic characteristics of object-oriented systems: information hiding and dynamic binding.
76. Utilizes the Unified Modeling Language (UML) and the Unified Process.
77. Characteristics of Object-Oriented Systems.
78. Concepts of Object-Oriented Programming: objects and classes.
79. Concepts of Object-Oriented Programming: abstraction.

80. Concepts of Object-Oriented Programming: polymorphism.
81. Concepts of Object-Oriented Programming: encapsulation, inheritance.
82. Concepts of Object-Oriented Programming: inheritance.
83. Architectural views of the system: functional (external), structural (static), behavioral (dynamic).
84. Benefits of Object-Oriented Systems Analysis and Design.
85. The Unified Process and its phases: inception phase.
86. The Unified Process and its phases: elaboration phase.
87. The Unified Process and its phases: construction phase.
88. The Unified Process and its phases: transition phase.
89. The Unified Process workflows: engineering workflow and supporting workflow.
90. The Unified Process. Engineering workflows: business-modeling.
91. The Unified Process. Engineering workflows: requirements.
92. The Unified Process. Engineering workflows: analysis.
93. The Unified Process. Engineering workflows: design.
94. The Unified Process. Engineering workflows: implementation.
95. The Unified Process. Engineering workflows: test.
96. The Unified Process. Engineering workflows: deployment workflows.
97. The Unified Process. Supporting workflows: project management.
98. The Unified Process. Supporting workflows: configuration and change management.
99. The Unified Process. Supporting workflows: environment workflow.

Compiled by:
Senior Teacher of the Department
“System and Application Programming”



Azizova Z.I.