Department of Computer Science and Engineering

***Course File***

**INTRODUCTION TO DATASCIENCE** (CS502PC)

**III B Tech I Semester**

**Academic Year: 2024-2025**

**S.SUNDEEP**

Assistant Professor, Department of CSE

****

**SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**

**(UGC Autonomous)**

(Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)

Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\)

**Vision of Institution - SDES**

* To develop this Institute as one of the premier & top-class institution in India.
* To be an academic institution in dynamic equilibrium with its social ecological and economic environment, striving continuously to excellence in education, research and technological service to nation.

**Mission of Institution- SDES**

* To provide high quality enterprising students with excellent technological skills.
* To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical and economic issues.
* To pursue research and disseminate research findings.
* To help in building national capabilities in science, technology, humanities, management, education and research.

**Department of Computer Science and Engineering**

**Vision**

* To become a pioneer in the field of Computer Science and allied Engineering for academic excellence.

**Mission**

* Interact with industry for professional development to meet the current industrial and societal needs.
* Offer quality education by implementing innovative teaching and learning practices.
* Promote and Involve in social, professional and leadership activities, for upliftment of quality of life.

**Program Educational Objectives (PEOs) for B.Tech(CSE)**

At the end of the program, the engineers will be able to:

PEO1: Establish as practicing professionals or researchers with continuous learning to solve problems in industry and society.

PEO2: Apply skills with mathematical, core engineering and contemporary technologies to analyze the requirements, prepare technical specifications, design and provide novel engineering solutions.

PEO3: Work as teams on multidisciplinary projects with leadership qualities, interpersonal, professional skills and ethical values.

**Program Outcomes (POs):** POs are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, analytical ability attitude and behavior that students acquire through the program.

The POs essentially indicate what the students can do from subject-wise knowledge acquired by them during the program. As such, POs define the professional profile of an engineering graduate.

**NBA has defined the following twelve POs for an engineering graduate. These are in line with the Graduate Attributes as defined by the Washington Accord:**

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Program Specific Outcomes (PSOs) for B.Tech(CSE)**

PSO1: Professional Skills: The ability to understand, analyse and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.

PSO2: Problem-Solving Skills: The ability to apply standard practices and strategies in software development using open-ended programming environments to deliver a quality product for business resources

PSO3: Sucessful Career and Entrepreneurship: The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur, enthusiasm for higher education, also include good manners and ethics for responsible, co-operative citizenship.

**Mapping of PEOs and POs/PSOs**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **PEO1** | √ | √ | √ | √ | √ |  |  |  | √ |  | √ |  | √ | √ | √ |
| **PEO2** |  |  | √ |  |  | √ | √ | √ |  | √ | √ |  | √ | √ | √ |
| **PEO3** |  |  |  |  | √ | √ |  |  | √ | √ | √ | √ |  | √ | √ |

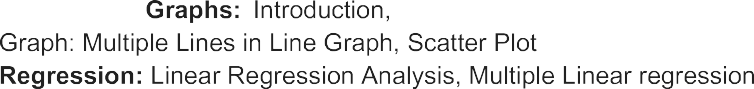
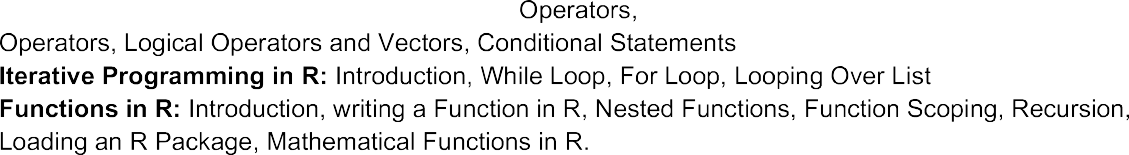
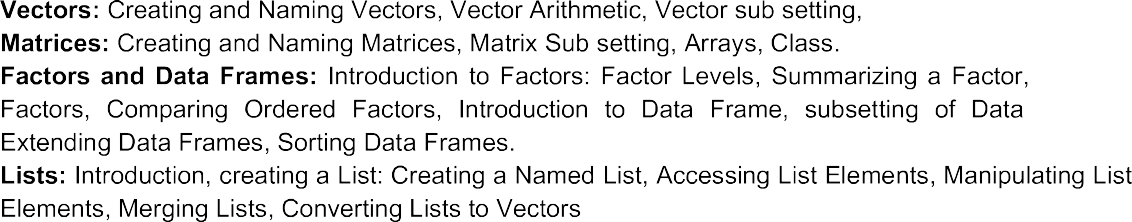
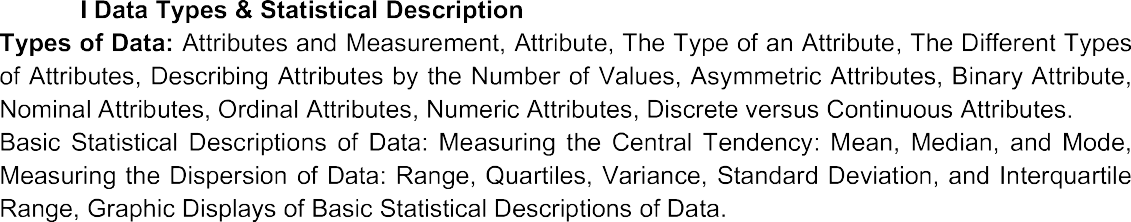






















**COURSE INFORMATION SHEET**

|  |  |
| --- | --- |
| PROGRAMME: B.Tech-Computer Science and Engineering | DEGREE: **UG** |
| COURSE: INTRODUCTION TO DATASCIENCE | YEAR III and SEMESTER: I CREDITS: 3 |
| COURSE CODE: AM512PE  REGULATION: R22 | COURSE TYPE: CORE |
| COURSE AREA/DOMAIN: | CONTACT HOURS: **3 hours / Week** |
| CORRESPONDING LAB COURSE CODE (IF ANY): - **NA** | LAB COURSE NAME: - **NA** |

|  |  |  |
| --- | --- | --- |
| **UNIT** | **DETAILS** | **HOURS** |
| I | . | 11 |
| II |  | 08 |
| III |  | 08 |
| IV | . | 09 |
| V |  | 09 |
| TOTAL HOURS | |  |
| Descriptive Tests | | -- |
| Topics beyond the Syllabus | | -- |
| Remedial classes | | -- |
| Tutorial classes | | -- |
| Total Number of Classes | | 43 |

**COURSE OBJECTIVE:**

 Learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration

 Understand the basic types of data and basic statistics

 Identify the importance of data reduction and data visualization techniques

**COURSE OUTCOMES:**

At the end of the course, the student will be able to:

CO1: Understand the fundamental concepts of Data Science, including statistical inference, data modeling, and basics of R programming.

CO2: Analyze different data types and apply statistical descriptions to summarize and interpret data.

CO3: Apply R programming skills to manage and manipulate data structures such as vectors, matrices, factors, data frames, and lists.

CO4: Implement conditional statements, control flow, and iterative programming in R to develop efficient data processing scripts.

CO5: Utilize R to create various charts and graphs, and perform linear and multiple regression analysis for data visualization and predictive modeling.

**Mapping of COs with POs and PSOs:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **COs** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 | 2 | 1 | 1 | 2 | - | - | - | - | 1 | - | 1 | 2 | 2 | - |
| **CO2** | 2 | 3 | 2 | 2 | 3 | - | - | - | - | 2 | - | 2 | 2 | 2 | - |
| **CO3** | 3 | 3 | 3 | 3 | 3 | 2 | - | - | - | 2 | 1 | 2 | 3 | 2 | 2 |
| **CO4** | 3 | 2 | 3 | 2 | 3 | 1 | - | - | - | 2 | 1 | 3 | 3 | 2 | 2 |
| **CO5** | 3 | 3 | 2 | 3 | 3 | 2 | 1 | - | - | 2 | 2 | 2 | 3 | 3 | 2 |

**Note: 0-No match, 1-Poor, 2-Light, 3-High**

**GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **DESCRIPTION** | **PROPOSED ACTIONS** |
| 1 | 1. **Real-World Applications and Case Studies.** 2. **Advanced Data Visualization and Additional Statistical Models.** 3. **Practical Data Preprocessing Techniques.** | 1.Case Studies and Examples. 2.Create Interactive Sessions.  3.Invite Industry Experts Conduct.  4.Hands-On Workshops. |

**PROPOSED ACTIONS: Topics beyond syllabus/assignment/industry visit/guest lecturer/NPTEL etc.**

**TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:**

|  |  |  |
| --- | --- | --- |
| 1 | **Data Wrangling Techniques**: | Essential methods for cleaning and transforming raw data into a usable format, applicable throughout the data analysis process. |
| 2 | **Advanced Statistical Metrics**: | Introduction to metrics like skewness, kurtosis, and z-scores, enhancing the statistical analysis capabilities taught in the course. |
| 3 | **Functional Programming Paradigms**: | Using functional programming concepts in R, including apply, lapply, and anonymous functions, to write more efficient and readable code. |
| 4 | **Interactive Visualizations with ggplot2:** | Creating advanced, interactive visualizations to better explore and present data insights, extending the basic charting techniques covered in the syllabus. |

**PROPOSED ACTIONS: Topics beyond syllabus/assignment/industry visit/guest lecturer/NPTEL etc.**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name of Book/Website** | **Links** |
| 1 | [DataCamp](https://www.datacamp.com/) | https://www.datacamp.com/ |
| 2 | [Coursera - Introduction to Data Science](https://www.coursera.org/specializations/data-science-python) | https://www.coursera.org/specializations/data-science-python |
| 3 | [Data Science Central](https://www.datasciencecentral.com/) | https://www.datasciencecentral.com/ |
| 4 | [Towards Data Science on Medium](https://towardsdatascience.com/) | https://towardsdatascience.com/ |
| 5 | ["Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani](https://www.statlearning.com/) | https://www.statlearning.com/ |
| 6 | "Python for Data Analysis" by Wes McKinney | https://wesmckinney.com/book/ |
| 7 | "Data Science from Scratch" by Joel Grus | https://www.oreilly.com/library/view/data-science-from/9781492041122/ |
| 8 | ["The Data Science Handbook" by Carl Shan, Henry Wang, William Chen, and Max Song](https://www.thedatasciencehandbook.com/) | https://www.thedatasciencehandbook.com/ |
| 9 | "Practical Statistics for Data Scientists" by Peter Bruce and Andrew Bruce | <https://www.researchgate.net/profile/Janine-Zitianellis/post> |

**DELIVERY/INSTRUCTIONAL METHODOLOGIES:**

|  |  |  |  |
| --- | --- | --- | --- |
| √ S/P | √ STUD. ASSIGNMENT | √ WEB RESOURCES |  |
| ☐ LCD/SMART BOARDS | √ STUD. SEMINARS | ☐ ADD-ON COURSES |  |

**ASSESSMENT METHODOLOGIES-DIRECT**

|  |  |  |  |
| --- | --- | --- | --- |
| √ ASSIGNMENTS | ☐ STUD. SEMINARS | √ TESTS/MODEL EXAMS | √ UNIV. EXAMINATION |
| ☐ STUD. LAB PRACTICES | ☐ STUD. VIVA | ☐ MINI/MAJOR PROJECTS | ☐ CERTIFICATIONS |
| ☐ ADD-ON COURSES | ☐ OTHERS |  |  |

**ASSESSMENT METHODOLOGIES-INDIRECT**

|  |  |
| --- | --- |
| √ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE) | √ STUDENT FEEDBACK ON FACULTY (TWICE) |
| ☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS | ☐ OTHERS |

**Prepared by Faculty Approved by HoD**

**Faculty Name HoDs Name**

|  |  |
| --- | --- |
|  | **SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**  **(UGC Autonomous)**  (Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)  Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\) |

**Department of Computer Science and Engineering**

**Lesson Plan**

Name of the Subject : Software Engineering

Academic Year : 2024-25

Subject Code : CS502PC Year & Semester: II-I

Name of the Faculty Member : **Faculty Name**

Class & section : II CSE A

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Date** | **Name of the Topic** | | **No. of Classes require** | | **Cumulative number of periods** | | **Teaching AID** |
|  | UNIT-I | | | | | | | |
| 1 | 12/10/2023 | Introduction to SE | | 01 | | 01 | | S/P |
| 2 | 13/10/2023 | The evolving role of software | | 01 | | 02 | | C&T, S/P |
| 3 | 20/10/2023 | Changing nature of software, software myths. | | 01 | | 03 | | C&T, S/P |
| 4 | 21/10/2023 | A Generic view of process | | 01 | | 04 | | C&T, S/P |
| 5 | 02/11/2023 | A layered technology | | 01 | | 05 | | S/P |
| 6 | 03/11/2023 | a process framework, CMMI | | 01 | | 06 | | S/P |
| 7 | 04/11/2023 | process assessment | | 01 | | 07 | | C&T, S/P |
| 8 | 09/11/2023 | personal and  team process models. | | 01 | | 08 | | S/P |
| 9 | 10/11/2023 | The waterfall model, incremental process models | | 01 | | 09 | | C&T, S/P |
| 10 | 16/11/2023 | evolutionary process models | | 01 | | 10 | | S/P |
| 11 | 17/11/2023 | The unified process. | | 01 | | 11 | | S/P |
|  | **UNIT-II** | | | | | | | |
| 12 | 18/11/2023 | Functional and non-functional requirements | | 01 | | 12 | | S/P |
| 13 | 23/11/2023 | user requirements, system  requirements | | 01 | | 13 | | S/P |
| 14 | 24/11/2023 | interface specification, the software requirements document. | | 01 | | 14 | | C&T, S/P |
| 15 | 25/11/2023 | Feasibility studies, requirements elicitation and analysis, | | 01 | | 15 | | S/P |
| 16 | 01/12/2023 | requirements validation, requirements management | | 01 | | 16 | | C&T, S/P |
| 17 | 02/12/2023 | Context models, behavioral models | | 01 | | 17 | | S/P |
| 18 | 07/12/2023 | data models, object models | | 01 | | 18 | | C&T |
| 19 | 08/12/2023 | structured methods. | | 01 | | 19 | | C&T |
|  | **UNIT-III** | | | | | | | |
| 20 | 21/12/2023 | design concepts, the design model | | 01 | | | 20 | C&T, S/P |
| 21 | 22/12/2023 | software architecture, data design | | 01 | | | 21 | S/P |
| 22 | 23/12/2023 | architectural styles and patterns | | 01 | | | 22 | C&T, S/P |
| 23 | 28/12/2023 | architectural design | | 01 | | | 23 | S/P |
| 24 | 29/12/2023 | conceptual model of UML | | 01 | | | 24 | C&T |
| 25 | 30/12/2023 | basic structural modeling | | 01 | | | 25 | C&T |
| 26 | 04/01/2024 | class diagrams, sequence  diagrams | | 01 | | | 26 | C&T, S/P |
| 27 | 05/01/2024 | collaboration diagrams, use case diagrams, component diagrams | | 01 | | | 27 | S/P |
|  | **UNIT-IV** | | | | | | | |
| 28 | 06/01/2024 | | Testing Strategies | | 01 | | 28 | S/P |
| 29 | 11/01/2024 | | software testing, test strategies for conventional software | | 01 | | 29 | S/P |
| 30 | 12/01/2024 | | black-box and white-box testing | | 01 | | 30 | S/P |
| 31 | 18/01/2024 | | system testing, the art of debugging | | 01 | | 31 | C&T, S/P |
| 32 | 19/01/2024 | | Software quality, metrics for analysis model | | 01 | | 32 | S/P |
| 33 | 20/01/2024 | | metrics for design model, metrics for source code | | 01 | | 33 | C&T, S/P |
| 34 | 25/01/2024 | | metrics for testing | | 01 | | 34 | C&T, S/P |
| 35 | 27/01/2024 | | metrics for maintenance | | 01 | | 35 | C&T |
|  | **UNIT-V** | | | | | | | |
| 36 | 01/02/2024 | | Software measurement, metrics for software quality | | 01 | | 36 | S/P |
| 37 | 02/02/2024 | | Reactive Vs proactive risk strategies | | 01 | | 37 | S/P |
| 38 | 03/02/2024 | | software risks, risk identification | | 01 | | 38 | S/P |
| 39 | 08/02/2024 | | Risk projection, risk refinement, RMMM, RMMM plan. | | 01 | | 39 | S/P |
| 40 | 09/02/2024 | | Quality concepts, software quality assurance | | 01 | | 40 | S/P |
| 41 | 10/02/2024 | | software reviews, formal technical reviews | | 01 | | 41 | S/P |
| 42 | 15/02/2024 | | statistical software quality assurance | | 01 | | 42 | S/P |
| 43 | 16/02/2024 | | software reliability, the ISO 9000 quality standards | | 01 | | 43 | S/P |
| 44 | 17/02/2024 | | Revision | | 01 | | 44 | C&T |
|  |  | | **Total number of classes** | | **44** | | **44** |  |

Teaching Methods:

C&T:-C&T; S/P:-Slides/PPT; Videos; SEM: Seminar; Demo; CHART; ET/GL: Expert Talk/Guest Lecture; QUIZ; CPS: Class room problem solving; GD:-Group discussion; RTCS: Real time case studies; JAR:-Journal article review; PD:-Poster design; OL:-Online lecture/Google class room; Industrial Visit (IV),

Assignment (ASG), Quiz/Puzzle (Q), Brain storming (BS), Think-Pair-Share (TPS), Certification(CERT), SIM: Simulation, P/G: Pledge/Greeting, Q/R: Quotes, references, LS: Literature Survey, RW: Report Writing, MM: Model making, PED: Professional/ethical dilemma, Coding, Activity/Event, FV: Filed Visit etc.

**MODEL LESSON PLAN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Name of the Topic** | **No. of Classes required** | **Cumulative number of periods** | **Teaching AID** |
| 1 | Course Objectives and Course Outcomes Explanation | 01 | 01 | Chalk & Talk |
| 2 | Unit-1 Evolving role of Software, Changing Nature of Software. | 01 | 02 | Chalk & Talk |
| 3 | Software Myths,Software engineering- A layered technology | 01 | 03 | PPT |
| 4 | **Tutorial class:** Software myths | 01 | 04 | Practice |
| 5 | A process framework,  The Capability Maturity Model Integration (CMMI) | 01 | 05 | Chalk & Talk |
| 6 | Process patterns, Process assessment | 01 | 06 | PPT |
| 7 | **Tutorial class:** The Capability Maturity Model Integration (CMMI) | 01 | 07 | Practice |
| 8 | Personal and team process models | 01 | 08 | PPT |
| 09 | The waterfall model. | 01 | 09 | Chalk & Talk |
| 10 | Incremental process models | 01 | 10 | PPT |
| 11 | Evolutionary process models | 01 | 11 | PPT |
| 12 | **Tutorial Class:** Incremental process models | 01 | 12 | Practice |
| 13 | The Unified process | 01 | 13 | Chalk & Talk |
|  | Unit -1 submission of student assignment |  |  | Student assignment |
| 14 | Functional and non-functional requirements | 01 | 14 | PPT |
| 15 | User requirements, System requirements | 01 | 15 | PPT |
| 16 | Interface specification, the software requirements document. | 01 | 16 | Chalk & Talk |
| 17 | **Tutorial Class:** The software requirements document. | 01 | 17 | Practice |
| 18 | Descriptive test - 1 | 01 | 18 | Test |
| 19 | Feasibility studies | 01 | 19 | Chalk & Talk |
| 20 | Requirements elicitation and analysis | 01 | 20 | PPT |
| 21 | Requirements validation, | 01 | 21 | PPT |
| 22 | **Tutorial Class**: Feasibility studies | 01 | 22 | Practice |
| 23 | Requirements management | 01 | 23 | PPT |
| 24 | Context Models, Behavioral models | 01 | 24 | Chalk & Talk |
| 25 | **Tutorial Class:** Context Models | 01 | 25 | Practice |
| 26 | Data models, Object models, Structured methods. | 01 | 26 | PPT |
|  | Unit -2 submission of student assignment |  |  | Student assignment |
| 27 | Design process and Design quality | 01 | 27 | Chalk & Talk |
| 28 | Design concepts | 01 | 28 | PPT |
| 29 | The design model | 01 | 29 | PPT |
| 30 | Software architecture | 01 | 30 | PPT |
| 31 | **Tutorial class:** The design model | 01 | 31 | Practice |
| 32 | Data design | 01 | 32 | PPT |
| 33 | Architectural styles and patterns | 01 | 33 | Chalk & Talk |
| 34 | Architectural Design, | 01 | 34 | Chalk & Talk |
| 35 | **Tutorial class:** Architectural styles and patterns | 01 | 35 | Practice |
| 36 | **TOPICS BEYOND SYLLABUS:** Software Architecture  Study the concepts, principles, methods, and best practices in software architecture. Different architectural styles, patterns | 02 | 37 | NPTEL |
| 37 | **GAPS IN THE SYLLABUS:**  UML Design: In the lab students are generating UML Diagrams, which is useful for the Project SRS Documentation. | 02 | 39 | Guest Lecture |
| 38 | Conceptual model of UML | 01 | 40 | Chalk & Talk |
| 39 | Basic Structural modeling | 01 | 41 | PPT |
| 40 | Class Diagrams | 01 | 42 | PPT |
| 41 | **Tutorial class:** Basic Structural modeling | 01 | 43 | Practice |
| 42 | Sequence diagram | 01 | 44 | PPT |
| 43 | Collaboration diagrams | 01 | 45 | Chalk & Talk |
| 44 | Use Case diagrams | 02 | 47 | Chalk & Talk |
| 45 | Component diagrams | 01 | 48 | Chalk & Talk |
| 46 | **Tutorial Class:** Collaboration diagrams | 01 | 49 | Practice |
|  | Unit -3 submission of student assignment |  |  | Student assignment |
| 47 | A strategic approach to software testing | 01 | 50 | Chalk & Talk |
| 48 | Test strategies for conventional software | 01 | 51 | PPT |
| 49 | Black-Box and White-Box testing | 01 | 52 | PPT |
| 50 | Validation testing , System testing | 01 | 53 | Chalk & Talk |
| 51 | **Tutorial Class**: Black-Box and White-Box testing | 01 | 54 | Practice |
| 52 | The art of Debugging | 01 | 55 | PPT |
| 53 | Software Quality | 01 | 56 | PPT |
| 54 | Metrics for Analysis Model | 01 | 57 | Chalk & Talk |
| 55 | Metrics for Design Model & Source Code | 01 | 58 | Chalk & Talk |
| 56 | **Tutorial Class:** Metrics for Analysis Model | 01 | 59 | Practice |
| 57 | Metrics for testing, Metrics for maintenance. | 02 | 61 | Chalk & Talk |
| 58 | **GAPS IN THE SYLLABUS:** Testing Tools | 01 | 62 | Guest Lecture |
| 59 | **Tutorial Class:** Metrics for testing | 01 | 63 | Chalk & Talk |
|  | Unit-4 submission of student assignment |  |  | Student assignment |
| 60 | Descriptive test - 2 | 01 | 64 | Test |
| 61 | Software Measurement | 01 | 65 | PPT |
| 62 | Metrics for Software Quality | 01 | 66 | PPT |
| 63 | Reactive vs. Proactive Risk strategies | 01 | 67 | PPT |
| 64 | software risks, Risk identification, Risk projection | 02 | 69 | PPT |
| 65 | Risk refinement, RMMM, RMMM Plan | 01 | 70 | PPT |
| 66 | **Tutorial class:** Risk Projections | 01 | 71 | Practice |
| 67 | Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews | 01 | 72 | Chalk & Talk |
| 68 | **Tutorial Class** : Quality concepts, Software quality assurance | 01 | 73 | Practice |
| 69 | Statistical Software quality Assurance | 01 | 74 | PPT |
| 70 | Software reliability | 01 | 75 | PPT |
| 71 | The ISO 9000 Quality standards | 01 | 76 | PPT |
| 72 | **Topic beyond the Syllabus**: Software Project Management  Introduction project management concepts, tools, and techniques | 01 | 77 | NPTEL |
| 73 | **Tutorial class:** Quality standards | 01 | 78 | Practice |
|  | Unit-5 submission of student assignment |  |  | Student assignment |
| 74 | Remedial class – practicing important questions from unit-1-2 | 01 | 79 | Practice |
| 75 | Remedial class – practicing important questions from unit 4-5 | 01 | 80 | Practice |
|  | **Tutorial Classes :16 Descriptive Tests :2**  **Classes for beyond syllabus :02 Remedial Classes/NPTL :05**  **TOTAL NUMBER OF CLASSES** | | | 80 |

**TEXT / REFERENCE BOOKS:**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **TEXT / REFERENCE BOOKS** | **BOOK TITLE/AUTHORS/PUBLICATION** |
|  | Text Book | Software Engineering, A practitioner’s Approach- Roger S. Pressman, 6th edition, Mc Graw Hill  International Edition. |
|  | Text Book | Software Engineering- Sommerville, 7th edition, Pearson Education. |
|  | Text Book | The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson,  Pearson Education. |
|  | Reference Book | Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley. |
|  | Reference Book | Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies. |
|  | Reference Book | Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education. |

**Signature of Faculty HoD-CSE**

|  |  |
| --- | --- |
|  | **SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**  **(UGC Autonomous)**  (Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)  Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\) |

**Lecture Plan with Blooms Taxonomy**

Name of the Subject : Software Engineering `

Subject Code : CS502PC

Name of the Faculty Member : **Faculty Name**

Class & section : II CSE A

**BTL- Blooms Taxonomy Level:**

Level 1- Remembering

Level 2- Understanding

Level 3 - Applying

Level 4 - Analyzing

Level 5 - Evaluating

Level 6 – Creating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cumulative Periods | Time(Min) | Topics | BTL | Teaching – Learning Method |
| 1 | 20 | Attendance; What is Software, Types of software | 1 | S/P |
| 30 | Introduction to Software Engineering | 1 | C&T, S/P |
| 2 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | The evolving role of software | 2 | S/P |
| 20 | Characteristics of Software | 2 | S/P |
| 3 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 10 | Changing nature of software | 2 | S/P |
| 10 | software myths. | 2 | C&T, S/P |
| 20 | Examples | 2 | C&T, S/P |
| 4 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | A Generic view of process | 1,2 | C&T, S/P |
| 5 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | A layered technology | 2 | S/P |
| 6 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | a process framework, CMMI | 2 | C&T, S/P |
| 7 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 30 | process assessment | 2 | S/P |
| 10 | Examples | 1,2 | S/P |
| 8 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | personal and  team process models. | 2 | S/P |
| 9 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | The waterfall mode | 2 | S/P |
| 20 | incremental process models | 2 | S/P |
| 10 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 30 | evolutionary process models | 1,2 | C&T |
| 10 | The unified process | 1,2 | S/P |
| 11 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 30 | Revision of unit 1 | 1,2 | S/P |
| 10 | Assigning some examples to group of students in order to understand the different types of software and advantages and disadvantages | 1 | S/P |
| **UNIT-II** | | | | |
| 12 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | Functional requirements | 2 | S/P |
| 20 | non-functional requirements | 2 | S/P |
| 13 | 10 | Attendance and Discussion of previous topics | 2 | S/P |
| 20 | user requirements | 1,2 | S/P |
| 20 | system  requirements | 2 | S/P |
| 14 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | interface specification | 2 | C&T, S/P |
| 20 | the software requirements document | 1,2 | C&T, S/P |
| 15 | 10 | Attendance and Discussion of previous topics | 1 | C&T, S/P |
| 20 | Feasibility studies | 2 | S/P |
| 20 | requirements elicitation and analysis | 1,2 | S/P |
| 16 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | requirements validation | 2 | C&T, S/P |
| 20 | requirements management | 1,2 | S/P |
| 17 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | Context models, behavioral models | 2 | S/P |
| 18 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | data models, object models | 3,4,5 | C&T, S/P |
| 19 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | structured methods. | 1,2 | C&T, S/P |
| **UNIT-III** | | | | |
| 20 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | design concepts, the design model | 3,4,5 | C&T, S/P |
| 21 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | software architecture, data design | 3,4,5 | S/P |
| 22 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | architectural styles and patterns | 3,4,5 | C&T, S/P |
| 23 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | architectural design | 3,4,5 | S/P |
| 24 | 10 | Attendance and Discussion of previous topics | 1 |  |
| 40 | conceptual model of UML | 1,2,3 | C&T, S/P |
| 25 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | basic structural modeling | 2 | C&T, S/P |
| 20 | Examples | 1,2 | C&T, S/P |
| 26 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | class diagrams, sequence  diagrams | 3,4,5 | C&T, S/P |
| 27 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | collaboration diagrams, use case diagrams, component diagrams | 1,2 | C&T, S/P |
| **UNIT-IV** | | | | |
| 28 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | Testing Strategies | 2 | S/P |
| 29 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | software testing, test strategies for conventional software | 2 | S/P |
| 30 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | black-box and white-box testing | 2 | S/P |
| 31 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | system testing, the art of debugging | 2 | C&T, S/P |
| 20 | Associate Memory | 2 | C&T, S/P |
| 32 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | Software quality | 2 | C&T, S/P |
| 20 | metrics for analysis model | 2 | S/P |
| 33 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | metrics for design model | 1,2 | C&T, S/P |
| 20 | metrics for source code | 1,2 | C&T, S/P |
| 34 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | metrics for testing | 1,2 | C&T, S/P |
| 35 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | metrics for maintenance | 1,2 | C&T |
| **UNIT-V** | | | | |
| 36 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | Software measurement | 2 | S/P |
| 10 | metrics for software quality | 1,2 | S/P |
| 37 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | Reactive Vs proactive risk strategies | 2 | S/P |
| 38 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | software risks | 2 | C&T, S/P |
| 20 | risk identification | 2 | S/P |
| 39 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 30 | Risk projection, risk refinement | 2 | C&T, S/P |
| 10 | RMMM, RMMM plan. | 2 | C&T, S/P |
| 40 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | Quality concepts, software quality assurance | 2 | S/P |
| 41 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | software reviews | 2 | S/P |
| 20 | formal technical reviews | 2 | S/P |
| 42 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | statistical software quality assurance | 2 | C&T, S/P |
| 43 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 40 | software reliability, the ISO 9000 quality standards | 1,2 | C&T |
| 44 | 10 | Attendance and Discussion of previous topics | 1 | S/P |
| 20 | Revision | 1,2 | C&T |
| 20 | Slip test | 1,2 | C&T |

Total Classes= 44

Tutorial Classes= --

Classes for Beyond Syllabus --,

Remedial Classes ---,

Descriptive Tests --,

Gaps in the syllabus ---

Total Number of Classes= 44

|  |  |
| --- | --- |
|  | **SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**  **(UGC Autonomous)**  (Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)  Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\) |

**Question Bank with Blooms Taxonomy Level (BTL)**

Academic Year : 2024-2025

Subject Name with code : Software Engineering (CS502PC)

Class : II CSE A & B

Name of the Faculty Member : Faculty Name

**Blooms Taxonomy Levels (BTL)**

1. Remembering
2. Understanding
3. Applying
4. Analyzing
5. Evaluating
6. Creating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.No. | Questions  (Select Questions from University question Bank and mention year in bracket or you may give own standard question with (new) in bracket) | | BTL level (Please mention L1 or L2 or etc...) | Course Outcome (Please mention CO1 or CO2 etc…) |
|  | **Unit - I** | |  |  |
|  | **Part – A (2 Marks )** | |  |  |
| 1 | Define Software Engineering and legacy software? | L-1 | | CO 1 | |
| 2 | List the fundamental activities of a software process? | L-1 | | CO 1 | |
| 3 | Discuss about the Evolving Role of the Software? | L-2 | | CO 1 | |
| 4 | Explain about personal and team process models? | L-2 | | CO 1 | |
| 5 | List all the umbrella activities in process framework? | L-1 | | CO 2 | |
| 6 | List the types of software myths? | L-1 | | CO 1 | |
| 7 | Explain the evolving role of software and explain changing nature of software in detail? | L-2 | | CO 1 | |
| 8 | Demonstrate all the applications of software? | L-1 | | CO 1 | |
| 9 | Explain process assessment with a neat diagram? | L-2 | | CO 1 | |
| 10 | Compare both personal and team process models? | L-1 | | CO 1 | |
|  | **Part – B (5 Marks )** |  | |  | |
| 1 | Explain different levels of CMMI model? | L-2 | | CO 1 | |
| 2 | Explain about all the software process models? | L-2 | | CO 1 | |
| 3 | Explain software development life cycle. Discuss various activities during SDLC? | L-2 | | CO 1 | |
| 4 | List the types of software myths?  Discuss the architecture of layered technology? | L-1 | | CO 1 | |
| 5 | Define software and explain the various characteristics of software? | | L-1 | CO 1 |
| 6 | Explain about Waterfall software process model with a neat diagram? | | L-2 | CO 1 |
| 7 | Explain about Incremental and RAD software process models with a neat diagram? | | L-2 | CO 1 |
| 8 | Explain about 3 types of Evolutionary software process models with a neat diagram? | | L-2 | CO 1 |
| 9 | Explain about Unified software process model with a neat diagram? | | L-2 | CO 1 |
| 10 | Explain “Software myth”? Discuss on various types of software myths and the true aspects of these myths | | L-2 | CO 1 |
|  | **Unit – II** | |  |  |
|  | **Part – A (2 Marks )** | |  |  |
| 1 | Explain the kinds of system requirements? | | L-2 | CO 2 |
| 2 | Describe the structure of software requirement document (SRS)? | | L-1 | CO 2 |
| 3 | Does feasibility study play an important role in requirement engineering process? | | L-2 | CO 2 |
| 4 | What are the activities of requirements elicitation and analysis? | | L-2 | CO 2 |
| 5 | Explain about behavioral models and object models with suitable examples? | | L-2 | CO 3 |
| 6 | List the various types of feasibility studies | | L-2 | CO 3 |
| 7 | Explain context models and Data models in briefly with examples? | | L-2 | CO 3 |
| 8 | Define interface specification? | | L-2 | CO 3 |
| 9 | Discuss the main characteristics of data model for requirement engineering? | | L-2 | CO 3 |
| 10 | Differentiate forward and backward engineering? | | L-2 | CO 3 |
|  | **Part – B (5 Marks )** | |  |  |
| 1 | What is the goal of requirements analysis phase? Give reasons why the requirements analysis phase is a difficult one? | | L-2 | CO 2 |
| 2 | Differentiate between functional and non-functional requirements? | | L-3 | CO 2 |
| 3 | Briefly explain about the requirements analysis? | |  |  |
| 4 | Give out an overview of various system models? | | L-3 | CO 3 |
| 5 | Explain about significance of feasibility study in detail? | | L-2 | CO 2 |
| 6 | Demonstrate class hierarchy for library by using interface specification? | |  |  |
| 7 |  | | L-3 | CO 3 |
| 8 | Discuss how requirements are felicitated and validated in software project? | | L-2 | CO 2 |
| 9 | Discuss about the requirements engineering process? | | L-2 | CO 2 |
| 10 | “The functional requirements specification of a system should be both complete and | |  |  |
|  | **Unit – III** | |  |  |
|  | **Part – A (2 Marks )** | |  |  |
| 1 | What is component? Explain in detail about component diagrams with an example? | | L-1 | CO 3 |
| 2 | Explain the basic Structural modeling of UML? | | L-2 | CO 3 |
| 3 | Differentiate between coupling and cohesion? How do they affect software design? | | L-3 | CO 3 |
| 4 | Define Sequence diagram? Explain the concept of Sequence diagram with example? | | L-1 | CO 3 |
| 5 | Explain the design process and design quality? | | L-2 | CO 3 |
| 6 | Define Collaboration diagram? Explain the concept of Collaboration diagram with example? | | L-1 | CO 3 |
| 7 | Explain about the design process in software development process? | | L-2 | CO 3 |
| 8 | Describe quality attributes and its guidelines? | | L-1 | CO 4 |
| 9 | Explain why design is important in design engineering? | | L-2 | CO 3 |
| 10 | Explain basic structural modeling in UML? | | L-3 | CO 3 |
|  | **Part – B (5 Marks )** | |  |  |
| 1 | Discuss the design principles that reduce the user’s memory load in user interface? | | L-2 | CO 3 |
| 2 | Explain Class diagram and Use case diagram? Draw the Class diagram and Use case diagram for any one example? | | L-3 | CO 3 |
| 3 | Write taxonomy of architectural styles and give a brief description of each style | | L-2 | CO 3 |
| 4 | Explain about Conceptual model of the UML with neat diagram? | | L-2 | CO 3 |
| 5 | Explain State chart diagram and Activity diagram? Draw the State chart diagram and Activity diagram for any one example? | | L-2 | CO 3 |
| 6 | Explain with a neat diagram of architectural design? | | L-2 | CO 3 |
| 7 | Discuss briefly the following fundamental concepts of software design:  i) Abstraction ii) Modularity iii) Information hiding. | | L-2 | CO 3 |
| 8 | Explain software architecture in a detail manner | | L-3 | CO 3 |
| 9 | Discuss architectural styles and patterns | | L-3 | CO 3 |
| 10 | What are the design concepts in software engineering | | L-2 | CO 3 |
|  | **Unit – IV** | |  |  |
|  | **Part – A (2 Marks )** | |  |  |
| 1 | Differentiate between Alpha and Beta testing? | | L-3 | CO 4 |
| 2 | What are the metrics used for software maintenance? Discuss? | | L-2 | CO 4 |
| 3 | Discuss the metrics used for Analysis Model? | | L-2 | CO 4 |
| 4 | Distinguish between verification and validation? | | L-3 | CO 4 |
| 5 | Discuss software quality factors? Discuss their relative importance? | | L-2 | CO 5 |
| 6 | Compare black box testing with white box testing? | | L-2 | CO 4 |
| 7 | Explain about the importance of test strategies for conventional software | | L-2 | CO 4 |
| 8 | Explain about the testing steps with a neat diagram? | | L-2 | CO 4 |
| 9 | Discuss the process of debugging? | | L-2 | CO 4 |
| 10 | What is software metric? Discuss about the product metric | | L-3 | CO 5 |
|  | **Part – B (5 Marks )** | |  |  |
| 1 | What is the need of software testing? What are its main objectives and principles? | | L-2 | CO 4 |
| 2 | Explain black box testing and white box testing in detail? | | L-3 | CO 4 |
| 3 | Explain the Test Strategies for Conventional Software?  What are the metrics used for Design Model? Discuss? | | L-2 | CO 4 |
| 4 | Discuss about strategic approach to Software Testing? | | L-2 | CO 4 |
| 5 | Briefly discuss about integration testing strategies? | | L-3 | CO 4 |
| 6 | Demonstrate art of debugging and a framework for product metrics? | | L-3 | CO 4 |
| 7 | Explain the metrics for software quality? | | L-3 | CO 5 |
| 8 | Define software testing. Explain different testing strategies? | | L-2 | CO 4 |
| 9 | Explain a strategic approach to software testing in software engineering? | | L-2 | CO 4 |
| 10 | Explain Validation Testing and System Testing? | | L-4 | CO 4 |
|  | **Unit – V** | |  |  |
| **Part – A (2 Marks )** | | | | |
| 1 | Define maintenance. What are the types of software maintenance? | | L-1 | CO 5 |
| 2 | Write a detailed note on ISO 9000 quality standards? | | L-1 | CO 5 |
| 3 | Discuss about reactive and proactive risk strategies? | | L-2 | CO 5 |
| 4 | Explain casual review and Formal Technical Reviews? | | L-2 | CO 5 |
| 5 | Explain the risk management paradigm with a neat diagram? | | L-2 | CO 5 |
| 6 | Explain about Quality concepts? | | L-3 | CO 5 |
| 7 | Explain software quality assurance | | L-2 | CO 5 |
| 8 | Explain about Software Reviews and formal technical reviews | | L-2 | CO 5 |
| 9 | Discuss software reliability? | | L-2 | CO 5 |
| 10 | Discuss about various types of risks. Explain about RMMM | | L-4 | CO 5 |
|  | **Part – B (5Marks )** | |  |  |
| 1 | Explain in detail about risk projection and RMMM plan? | | L-2 | CO 5 |
| 2 | List the major risks in a software project. What are the major ways to abate the risk of cost and schedule overruns? | | L-1 | CO 5 |
| 3 | With a neat diagram explain the risk management paradigm? | | L-2 | CO 5 |
| 4 | Discuss software quality assurance. Explain the statistical software quality? | | L-2 | CO 5 |
| 5 | Discuss any four useful indicators for software quality in SE? | | L-3 | CO 5 |
| 6 | Explain Software measurement, metrics for software quality? | | L-2 | CO 5 |
| 7 | Elaborate the concepts of Risk management Reactive vs Proactive Risk Strategies? | | L-3 | CO 5 |
| 8 | Explain the activities of software quality assurance group to assist the software team in achieving high quality. | | L-3 | CO 5 |
| 9 | How the risk will be identified and Explain the RMMM Plan in brief? | | L-2 | CO 5 |
| 10 | Explain software risks? Explain risk identification? | | L-2 | CO 5 |

Signature of the Course Instructor

Name:

Verified by

1. Course Coordinator :
2. Module Coordinator :
3. Department coordination committee Head :

|  |  |
| --- | --- |
|  | **SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**  **(UGC Autonomous)**  (Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)  Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\) |

**Assignment Questions with Blooms Taxonomy Level (BTL)**

Academic Year : 2023-2024

Subject Name with code : Software Engineering (CS502PC)

Class& Section : II CSE A & B

Name of the Faculty Member : Faculty Name

**Blooms Taxonomy Levels (BTL)**

1. Remembering
2. Understanding
3. Applying
4. Analyzing
5. Evaluating
6. Creating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.No. | Questions | | BTL level (Please mention L1 or L2 or etc...) | Course Outcome (Please mention CO1 or CO2 etc…) |
|  | **Unit - I** | |  |  |
| 1 | Explain software process? What is need of software process improvement? Discuss capability and maturity models? | L-2 | | CO 1 | |
| 2 | Give an overview of unified process model with neat diagram? | L-2 | | CO 1 | |
| 3 | a) Explain the various software myths  b) Explain the working of process models | L-2 | | CO 1 | |
| 4 | Explain the following: i) Water fall model II) Incremental process models (Incremental model and RAD Model) iii) Evolutionary Process Models (Spiral Model, Prototyping, Concurrent Development Model). | L-2 | | CO 1 | |
| 5 | What is Software Development Life Cycle and explain all the phases? | L-2 | | CO 1 | |
|  | **Unit – II** | |  |  |
| 1 | Explain the structure of Software Requirements document. and What are the feasibility studies for requirements engineering process? | | L-2 | CO 2 |
| 2 | Explain Functional and non-functional requirements in Software Engineering? | | L-2 | CO 2 |
| 3 | Explain interface specification and explain the software requirements document? | | L-2 | CO 2 |
| 4 | Explain Context models, behavioral models, data models, object models, structured methods in detail | | L-2 | CO 3 |
| 5 | Explain requirements validation and requirements management? | | L-2 | CO 2 |
|  | **Unit – III** | |  |  |
| 1 | Explain the conceptual and structural model of UML with a neat diagram? | | L-2 | CO 3 |
| 2 | Explain the design of use case-based components or use case diagram with example? | | L-2 | CO 3 |
| 3 | Explain the design of class-based components or class diagram with example? | | L-2 | CO 3 |
| 4 | Explain the design of sequence and collaboration diagram with example? | | L-2 | CO 3 |
| 5 | Explain the software architecture in Software Engineering and Write a brief essay on design concepts in design engineering? | | L-2 | CO 3 |
|  | **Unit – IV** | |  |  |
| 1 | Illustrate the need of software testing? What are its main objectives and principles? | | L-3 | CO 4 |
| 2 | Explain the Test Strategies for Conventional Software and What are the metrics used for Design Model? | | L-2 | CO 4 |
| 3 | Discuss about strategic approach to Software Testing and Briefly discuss about integration testing strategies? | | L-2 | CO 4 |
| 4 | Compare and contrast black box testing and white box testing techniques in detail? | | L-3 | CO 4 |
| 5 | Differentiate between Alpha and Beta testing?  Distinguish between verification and validation? | | L-3 | CO 4 |
|  | **Unit – V** | |  |  |
| 1 | Explain in detail ISO 9000 quality standards? | | L-3 | CO 5 |
| 2 | Explain about RMMM Plan and quality concepts? | | L-2 | CO 5 |
| 3 | Discuss software quality assurance. Explain the statistical software quality? | | L-3 | CO 5 |
| 4 | Elaborate the concepts of Risk management Reactive vs Proactive Risk strategies? | | L-3 | CO 5 |
| 5 | Explain with a neat diagram for the risk management paradigm? | | L-2 | CO 5 |

|  |  |
| --- | --- |
|  | **SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**  **(UGC Autonomous)**  (Established by Vyjayanthi Educational Society, Accredited by NAAC,NBA(ECE&CIV),approved by AICTE and affiliated to JNTUH)  Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist – 501510. [www.sreedattha.ac.in/sdes/](file:///E:\brecw%20backup%201.8.24\Backup%20of%20downloads%2030.7.24\www.sreedattha.ac.in\sdes\) |

**MULTIPLE CHOICE QUESTIONS**

**UNIT-1**

1. RAD stands for  
   a) Relative Application Development  
   b) Rapid Application Development  
   c) Rapid Application Document

d)Relative Application Document

1. Which one of the following models is not suitable for accommodating any change?  
   a) Build & Fix Model  
   b) Prototyping Model  
   c) RAD Model  
   d) Waterfall Model
2. Which one of the following is not a phase of Prototyping Model?  
   a) Quick Design  
   b) Coding  
   c) Prototype Refinement  
   d) Engineer Product
3. What is the major drawback of using RAD Model?  
   a) Highly specialized & skilled developers/designers are required.  
   b) Increases re-usability of components.  
   c) Encourages customer/client feedback.  
   d) Both a & c.
4. SDLC stands for  
   a) Software Development Life Cycle  
   b) System Development Life cycle  
   c) Software Design Life Cycle  
   d) System Design Life Cycle
5. Which model can be selected if user is involved in all the phases of SDLC?  
   a) Waterfall Model  
   b) Prototyping Model  
   c) RAD Model  
   d) both b & c
6. What is Software?  
   a) Software is set of programs  
   b) Software is documentation and configuration of data  
   c) Software is set of programs, documentation & configuration of data  
   d) Software is an instruction.
7. Which of these does not account for software failure?  
   a) Increasing Demand  
   b) Low expectation  
   c) Increasing Supply  
   d) Less reliable and expensive
8. What are attributes of good software?  
   a) Software maintainability  
   b) Software functionality  
   c) Software development  
   d) Software maintainability & functionality
9. Which of these software engineering activities are not a part of software processes?  
   a) Software dependence  
   b) Software development  
   c) Software validation  
   d) Software specification
10. Which of these is incorrect?  
    a) Software engineering belongs to Computer science  
    b) Software engineering is a part of more general form of System Engineering  
    c) Computer science belongs to Software engineering  
    d) Software engineering is concerned with the practicalities of developing and delivering useful software
11. The Incremental Model is a result of combination of elements of which two models?  
    a) Build & FIX Model & Waterfall Model  
    b) Linear Model & RAD Model  
    c) Linear Model & Prototyping Model  
    d) Waterfall Model & RAD Model
12. What is the major advantage of using Incremental Model?  
    a) Customer can respond to each increment  
    b) Easier to test and debug  
    c) It is used when there is a need to get a product to the market early  
    d) Easier to test and debug & It is used when there is a need to get a product to the market early
13. How is Incremental Model different from Spiral Model?  
    a) Progress can be measured for Incremental Model  
    b) Changing requirements can be accommodated in Incremental Model  
    c) Users can see the system early in Incremental Model  
    d) All of the mentioned
14. If you were to create client/server applications, which model would you go for?  
    a) WINWIN Spiral Model  
    b) Spiral Model  
    c) Concurrent Model  
    d) Incremental Model
15. The importance of software design can be summarized in a single word which is:  
    a) Efficiency  
    b) Accuracy  
    c) Quality  
    d) Complexity
16. Which two models doesn’t allow defining requirements early in the cycle?  
    a) Waterfall & RAD  
    b) Prototyping & Spiral  
    c) Prototyping & RAD  
    d) Waterfall & Spiral
17. Which of the following life cycle model can be chosen if the development team has less experience on similar projects?  
    a) Spiral  
    b) Waterfall  
    c) RAD  
    d) Iterative Enhancement Model
18. Which two of the following models will not be able to give the desired outcome if user’s participation is not involved?  
    a) Waterfall & Spiral  
    b) RAD & Spiral  
    c) RAD & Waterfall  
    d) RAD & Prototyping
19. Which of the following is not a maturity level in CMM?  
    a) Design  
    b) Repeatable  
    c) Managed  
    d) Optimizing

**UNIT -2**

1. UML interfaces are used to:  
   a) specify required services for types of objects**.**b) program in Java, but not in C++ or Smalltalk.  
   c) define executable logic to reuse across classes.  
   d) define an API for all classes.
2. Select the developer specific requirement?  
   a) Portability  
   b) Maintainability  
   c) Availability  
   d) Both a and b
3. Which one of the following is not a step of requirement engineering?  
   a) elicitation  
   b) design  
   c) analysis  
   d) documentation
4. Which is the first step in the software development life cycle?  
   a) Analysis  
   b) Design  
   c) Problem/Opportunity Identification  
   d) Development and Documentation
5. The user system requirements are the parts of which document?  
   a) SDD  
   b) SRS  
   c) DDD  
   d)DSD
6. Which is one of the most important stakeholders from the following?  
   a) Entry level personnel  
   b) Middle level stakeholder  
   c) Managers  
   d) Users of the software
7. In the Analysis phase, the development of the \_\_\_\_\_\_\_\_\_\_\_\_ occurs, which is a clear statement of the goals and objectives of the project.  
   a) documentation  
   b) flowchart  
   c) program specification  
   d) design
8. Which one of the following is a requirement that fits in a developer’s module?  
   a) Availability  
   b) Testability  
   c) Usability  
   d) Flexibility
9. “Consider a system where, a heat sensor detects an intrusion and alerts the security company.” What kind of a requirement the system is providing?  
   a) Functional  
   b) Non-Functional  
   c) Known Requirement
10. What are the four dimensions of Dependability?  
    a) Usability, Reliability, Security, Flexibility  
    b) Availability, Reliability, Maintainability, Security  
    c) Availability, Reliability, Security, Safety  
    d) Security, Safety, Testability, Usability
11. What is the first step of requirement elicitation?  
    a) Identifying Stakeholder  
    b) Listing out Requirements  
    c) Requirements Gathering  
    d) Requirement Testing
12. Interaction Diagram is a combined term for  
    a) Sequence Diagram + Collaboration Diagram  
    b) Activity Diagram + State Chart Diagram  
    c) Deployment Diagram + Collaboration Diagram  
    d) State Chart Diagram
13. What are the types of requirement in Quality Function Deployment (QFD)?  
    a) Known, Unknown, Undreamed  
    b) User, Developer  
    c) Functional, Non-Functional  
    d) Normal, Expected, Exciting
14. What kind of approach was introduced for elicitation and modeling to give a functional view of the system?  
    a) Object Oriented Design (by Booch)  
    b) Use Cases (by Jacobson)  
    c) Fusion (by Coleman)  
    d) Object Modeling Technique (by Rambaugh)
15. Which of the following diagram is time oriented?  
    a) Collaboration  
    b) Sequence  
    c) Activity

d)Usecase

1. Debugging is:  
   a) creating program code  
   b) finding and correcting errors in the program code  
   c) identifying the task to be computerized  
   d) creating the algorithm
2. How many Scenarios are there in elicitation activities?  
   a) One  
   b) Two  
   c) Three  
   d) Four
3. Which of the following requires design control measures, such as holding and recording design reviews and qualification tests?  
   a) CMM  
   b) ISO 9001  
   c) ISO 9000-3  
   d) CMMI
4. The CMM emphasizes  
   a) continuous process improvement  
   b) the need to record information  
   c) the need to accept quality system  
   d) both a and b
5. The primary objective of formal technical reviews is to find \_\_\_\_\_\_\_\_\_ during the process so that they do not become defects after release of the software.  
   a) errors  
   b) equivalent faults  
   c) failure cause  
   d) None of the mentioned

**UNIT 3**

1. Which is the first step in the software development life cycle?  
   a) Analysis  
   b) Design  
   c) Problem/Opportunity Identification  
   d) Development and Documentation
2. Which tool is use for structured designing?  
   a) Program flowchart  
   b) Structure chart  
   c) Data-flow diagram  
   d) Module
3. In the Analysis phase, the development of the \_\_\_\_\_\_\_\_\_\_\_\_ occurs, which is a clear statement of the goals and objectives of the project.  
   a) documentation  
   b) flowchart  
   c) program specification  
   d) design
4. Actual programming of software code is done during the \_\_\_\_\_\_\_\_\_\_\_\_ step in the SDLC.  
   a) Maintenance and Evaluation  
   b) Design  
   c) Analysis  
   d) Development and Documentation
5. Who designs and implement database structures.  
   a) Programmers  
   b) Project managers  
   c) Technical writers  
   d) Database administrators
6. . \_\_\_\_\_\_\_\_\_\_\_\_ is the process of translating a task into a series of commands that a computer will use to perform that task.  
   a) Project design  
   b) Installation  
   c) Systems analysis  
   d) Programming
7. Debugging is:  
   a) creating program code  
   b) finding and correcting errors in the program code  
   c) identifying the task to be computerized  
   d) creating the algorithm
8. What is testing process’ first goal?

a) Bug prevention  
b) Testing  
c) Execution  
d) Analyses

1. The importance of software design can be summarized in a single word which is:  
   a) Efficiency  
   b) Accuracy  
   c) Quality  
   d) Complexity
2. Cohesion is a qualitative indication of the degree to which a module  
   a) can be written more compactly  
   b) focuses on just one thing  
   c) is able to complete its function in a timely manner  
   d) is connected to other modules and the outside world
3. Coupling is a qualitative indication of the degree to which a module  
   a) can be written more compactly  
   b) focuses on just one thing  
   c) is able to complete its function in a timely manner  
   d) is connected to other modules and the outside world
4. At Conceptual level Class diagrams should include  
   a) operations only  
   b) attributes only  
   c) both (a) and (b)
5. Cyclomatic Complexity method comes under which testing method.  
   a) Yellow box  
   b) White box  
   c) Gray box  
   d) Black box
6. Constraints can be represented in UML by  
   a) {text}  
   b) [text] c) constraint  
   d) None of the mentioned
7. What is an object?  
   a) An object is an instance of a class.  
   b) An object includes encapsulation of data  
   c) An object is not an instance of a class
8. What is an abstract class?  
   a) A class that has direct instances, but whose descendants may have direct instances.  
   b) A class that has direct instances, but whose descendants may not have direct instances.  
   c) A class that has no direct instances, but whose descendants may have direct instances.
9. Which is a black box testing technique appropriate to all levels of testing?  
   a) Acceptance testing  
   b) Regression testing  
   c) Equivalence partitioning  
   d) Quality assurance
10. Software mistakes during coding are known as

a) errors  
b) failures  
c) bugs  
d) defects

1. What is normally considered as an adjunct to the coding step  
   a) Integration testing  
   b) Unit testing  
   c) Completion of Testing  
   d) Regression Testing
2. Which diagram in UML shows a complete or partial view of the structure of a modeled system at a specific time?  
   a) Sequence Diagram  
   b) Collaboration Diagram  
   c) Class Diagram  
   d) Object Diagram

**UNIT -4**

1. Software Debugging is a set of activities that can be planned in advance and conducted systematically.  
   a) True  
   **b) False**
2. Which of the following is not a software testing generic characteristics?  
   a) Different testing techniques are appropriate at different points in time  
   b) Testing is conducted by the developer of the software or an independent test group  
   c) Testing and debugging are different activities, but debugging must be accommodated in any testing strategy  
   d) None of the mentioned
3. ITG stands for  
   a) instantaneous test group  
   b) integration testing group  
   c) individual testing group  
   d) independent test group
4. By collecting \_\_\_\_\_\_\_\_ during software testing, it is possible to develop meaningful guidelines to halt the testing process.  
   a) Failure intensity  
   b) Testing time  
   c) Metrics  
   d) All of the mentioned
5. Effective testing will reduce \_\_\_\_\_\_\_ cost.  
   a) maintenance  
   b) design  
   c) coding  
   d) documentation
6. Test cases should uncover errors like  
   a) Nonexistent loop termination  
   b) Comparison of different data types  
   c) Incorrect logical operators or precedence  
   d) All of the mentioned
7. Which of the following errors should not be tested when error handling is evaluated?  
   a) Error description is unintelligibleb) Error noted does not correspond to error encountered  
   c) Error condition causes system intervention prior to error handling  
   d) Error description provide enough information to assist in the location of the cause of the error
8. What is normally considered as an adjunct to the coding step  
   a) Integration testing  
   b) Unit testingc) Completion of Testing  
   d) Regression Testing
9. In which testing level the focus is on customer usage?  
   a) Alpha Testing  
   b) Beta Testing  
   c) Validation Testing  
   d) Both Alpha and Beta
10. What is testing process’ first goal?  
    a) Bug preventionb) Testing  
    c) Execution  
    d) Analyses
11. Software mistakes during coding are known as  
    a) errors  
    b) failures  
    c) bugs  
    d) defects
12. Name an evaluation technique to assess the quality of test cases.  
    a) Mutation analysisb) Validation  
    c) Verification  
    d) Performance analysis
13. Which of the following is not included in External failure costs?  
    a) testing  
    b) help line support  
    c) warranty work  
    d) complaint resolution
14. Cyclomatic Complexity method comes under which testing method.  
    a) Yellow box  
    b) White box  
    c) Gray box  
    d) Black box
15. Which is a black box testing technique appropriate to all levels of testing?  
    a) Acceptance testing  
    b) Regression testing  
    c) Equivalence partitioningd) Quality assurance
16. Which of the following is the way of ensuring that the tests are actually testing code?  
    a) Control structure testing  
    b) Complex path testing  
    c) Code coverage  
    d) Quality assurance of software
17. Effective testing will reduce \_\_\_\_\_\_\_ cost.  
    a) maintenance  
    b) design  
    c) coding  
    d) documentation
18. Which of the following does not affect the software quality and organizational performance?  
    a) Market  
    b) Product  
    c) Technology  
    d) People
19. The intent of project metrics is:  
    a) minimization of development schedule  
    b) for strategic purposes  
    c) assessing project quality on ongoing basis  
    d) both a and c
20. Which of the following is not a direct measure of SE process?  
    a) Efficiency  
    b) Cost  
    c) Effort Applied  
    d) All of the mentioned

**UNIT-5**

1. Which of the following risk is the failure of a purchased component to perform as expected?  
   a) Product risk  
   b) Project risk  
   c) Business risk  
   d) Programming risk
2. Which one is not a risk management activity?  
   a) Risk assessment  
   b) Risk generation  
   c) Risk control  
   d) None of the mentioned
3. What is the product of the probability of incurring a loss due to the risk and the potential magnitude of that loss?  
   a) Risk exposure  
   b) Risk prioritization  
   c) Risk analysis  
   d) All of the mentioned
4. What threatens the quality and timeliness of the software to be produced?  
   a) Known risks  
   b) Business risks  
   c) Project risks  
   d) Technical risks
5. What threatens the viability of the software to be built?  
   a) Known risks  
   b) Business risks  
   c) Project risks  
   d) Technical risks
6. Which of the following is a systematic attempt to specify threats to the project plan?  
   a) Risk identification  
   b) Performance risk  
   c) Support risk  
   d) Risk projection
7. Which risks are associated with the overall size of the software to be built or modified?  
   a) Business impact risks  
   b) Process definition risks  
   c) Product size risks  
   d) Development environment risks\
8. Which risks are associated with constraints imposed by management or the marketplace?  
   **a) Business impact risks**b) Process definition risks  
   c) Product size risks  
   d) Development environment risks
9. Which of the following term is best defined by the statement:”the degree of uncertainty that the product will meet its requirements and be fit for its intended use.”?  
   **a**) Performance risk  
   b) Cost risk  
   c) Support risk  
   d) Schedule risk
10. Which of the following risk is the failure of a purchased component to perform as expected?  
    a) Product risk  
    b) Project risk  
    c) Business risk  
    d) Programming risk
11. Which of the following term is best defined by the statement: “There will be a change of organizational management with different priorities.”?  
    a) Staff turnover  
    b) Technology change  
    c) Management changed) Product competition
12. Which of the following term is best defined by the statement: “The underlying technology on which the system is built is superseded by new technology.”?  
    a) Technology changeb) Product competition  
    c) Requirements change  
    d) None of the mentioned
13. What assess the risk and your plans for risk mitigation and revise these when you learn more about the risk?  
    a) Risk monitoring  
    b) Risk planning  
    c) Risk analysis  
    d) Risk identification
14. Which of the following risks are derived from the organizational environment where the software is being developed?  
    a) People risks  
    b) Technology risks  
    c) Estimation risks  
    **d) Organizational risks**
15. Which of the following risks are derived from the software or hardware technologies that are used to develop the system?  
    a) Managerial risks  
    b) Technology risks  
    c) Estimation risks  
    d) Organizational risks
16. Software Debugging is a set of activities that can be planned in advance and conducted systematically.  
    a) True  
    b) False
17. Which requirements are the foundation from which quality is measured?  
    a) Hardware  
    b) Software  
    c) Programmers  
    d) None of the mentioned
18. Which of the following is not a SQA plan for a project?  
    a) evaluations to be performed  
    b) amount of technical work  
    c) audits and reviews to be performed  
    d) documents to be produced by the SQA group
19. Degree to which design specifications are followed in manufacturing the product is called  
    a) Quality Control  
    b) Quality of conformancec) Quality Assurance  
    d) Quality of Acceptance

20. Which of the following is not included in External failure costs?  
a) testing  
b) help line support

c) warranty work  
d) complaint resolution

**Answers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNIT-I** | **UNIT-II** | **UNIT-III** | **UNIT-IV** | **UNIT-V** |
| **1. B** | **1. A** | **1. C** | **1. B** | **1. A** |
| **2. D** | **2. D** | **2. B** | **2. A** | **2. B** |
| **3. B** | **3. B** | **3. C** | **3. D** | **3. A** |
| **4. D** | **4. C** | **4. D** | **4. C** | **4. D** |
| **5. A** | **5. B** | **5. D** | **5. C** | **5. B** |
| **6. C** | **6. D** | **6. D** | **6. A** | **6. D** |
| **7. C** | **7. C** | **7. B** | **7. A** | **7. C** |
| **8. C** | **8. B** | **8. A** | **8. B** | **8. A** |
| **9. D** | **9. A** | **9. C** | **9. D** | **9. A** |
| **10. A** | **10. C** | **10. B** | **10. A** | **10. A** |
| **11. C** | **11. A** | **11. D** | **11. C** | **11. C** |
| **12. C** | **12. A** | **12. B** | **12. A** | **12. A** |
| **13. D** | **13. D** | **13. B** | **13. A** | **13. A** |
| **14. A** | **14. B** | **14. A** | **14. B** | **14. D** |
| **15. C** | **15. B** | **15. A** | **15. C** | **15. B** |
| **16. C** | **16. B** | **16. C** | **16. D** | **16. B** |
| **17. B** | **17. D** | **17. C** | **17. C** | **17. B** |
| **18. A** | **18. C** | **18. C** | **18. A** | **18. B** |
| **19. D** | **19. A** | **19. B** | **19. D** | **19. B** |
| **20. C** | **20. A** | **20. D** | **20. A** | **20. A** |