**NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD**

**DEPARTMENT OF ENGINEERING – BSSE PROGRAM**

**SOFTWARE CONSTRUCTION & DEVELOPMENT – COURSE OUTLINE**

|  |  |
| --- | --- |
| **Credit Hours:** | 4(3-1) |
| **Pre-requisites:** | Software Design & Architecture |
| **Instructor:** | Dr. Uzair Iqbal |
| **Content:** | Apply a wide variety of software construction techniques and modeling, including clean code concepts, state-based and table-driven approaches to low-level design of software, Design simple languages and protocols suitable for a variety of applications, Generate code for simple languages and protocols using suitable tools, Create simple formal specifications of low-level software modules, check the validity of these specifications, and generate code from the specifications using appropriate tools, Design simple concurrent software, Analyze software to improve its efficiency, reliability, and maintainability |
| **Objectives:** | At the end of the course the students will be able to:   1. Understand the bad and good code practices. 2. Understand the Concepts of UML modeling in software development. 3. Design the behavioral modeling through software metrics 4. Compile the findings of UML modeling. |
| **Textbook:** | 1. Code Complete 2nd edition: A practical handbook of software construction, published by Microsoft Press, 2004 2. Sams Teach Yourself UML in 24 Hours, Third Edition.Publisher(s): Sams ISBN: 067232640X 3. Software Engineering by Ian Sommerville, 8th edition, Addison & Wesley. 2006 |
| **Reference Books:** | 1. Object-Oriented Software Construction, by Bertrand Meyer, Second Edition, Published by, Prentice Hall in 1997 |

|  |  |  |
| --- | --- | --- |
| **Week No.** | **Topics** | **Reference** |
| 1 | Course Outline, Assessment Criteria ,Introduction to software development aspects | Chapter 1  (Book3) |
| 2 | STATIC Behavioral model VS. DYNAMIC behavioral model.  Practice on different Case studies | External Source |
| 3 | Clean Code vs Bad code**,** Writing clean & good code practice | Chapter 1  (Book 1) |
| 4 | REGRESSIVE Monitoring on clean code , CODE REVIEW, SUMMARY  REGRESSIVE Monitoring on bad code | Chapter 2  (Book 1) |
| 5 | SYSTEM ENGINEERING, Context, Major Principles  Refactoring through UML models( Use case and Sequence Diagrams) | Chapter 16  (Book 1) |
| 6 | Managing the System Evolution, Summary (How it helps in Software Construction). System modeling with requirements | Hour 18  (Book 2) |
| 7 | ANTICIPATING CHANGE, CONSTRUCTION FOR VERIFICATION, White box and black box software development processes | External resource |
| 8 | STANDARDS IN CONSTRUCTION, Exercises and Revision | Chapter 11  (Book 1) |
| 9 | SYSTEMS MODELS IN CONTEXT OF SOFTWARE CONSTRUCTION, Synchronization and Currency in UML modeling | Chapter 6, External resource |
| 10 | Role of Specification in Software Construction: Introduction: Why Specifications, Behavioral Equivalence, Structure, Types, Example, Summary. Introduction of Petri Nets | External resource |
| 11 | Using State Machines for Analysis and Design; understanding through examples. Petri Nets vs State machine diagram | Hour 8( Book 2),  External resource |
| 12 | Petri Nets exercises. Introduction to Formatting in Clean Code | Chapter 5  (Book 1) |
| 13 | Introduction Vertical Formatting, vertical density, vertical openness  Introduction of horizontal formatting and horizontal openness | Chapter 5  (Book 1) |
| 14 | Error handling core keywords and discussion .Interrupt region and expansion region through state machine diagram | Chapter 3  (Book 1)  External resource |
| 15 | Formation of refactoring through UML models**.** Introduction of TDD and Exercises | Chapter 9,11  (Book 1) |
| 16 | Formation of different Finite Automata**, Transition tables** .Revision | External resource |