|  |  |  |
| --- | --- | --- |
| **Department: Computer Science** | | |
| **Module Title: System Development** | **Module Number: 08 Module** | **CP: 23** |
| **Course Title: Object Oriented Software Engineering** | **Course Code: SEng3092** | **Course CP: 5** |

**Course Description:**

Covers O-O concepts, tools, development life cycle, problem solving, modeling, analysis, and design, while utilizing UML (Unified Modeling Language) for O-O modeling. UML has become the standard notation for modeling O-O systems and is being embraced by major software developers like Microsoft and Oracle.

**Learning Outcomes**: On successful completion of this module the learner will be able to

* Describe in detail the theory, concepts and methods pertaining to the Unified Modeling Language (UML). 
* Create requirements using use case modeling concepts. 
* Demonstrate conceptual and technical skills in the analysis, design and implementation of a software system using Object Oriented Concepts. 
* Employ tools and techniques for Object Oriented Software Engineering,  Demonstrate an ability to adapt and solve problems in software development activities from specification to testing individually and as part of a team.

**Prerequisites**: -Fundamental of software engineering

**Required Texts**: Brahmin, Ali (1999), Object oriented System development, Mc-Graw Hill, USA.

**Teaching Learning Methods:**

The learning–teaching methodology will be student-centered with appropriate guidance of

instructor/s during the students’ activities .There will be Lecture, Demonstrations, Tutorials,

Reading assignments and Group Discussions as well as lab work

**Assessment Methods:** As stated in the University Legislation

**Schedule:**

The following is an outline of the order in which syllabus contents will be covered. The exact dates and due dates for assignments and exams can be found on the class calendar and are subject to change with notice.

|  |  |  |
| --- | --- | --- |
| **Week** | **Lecture Topics** | **Reading**  **Assignment** |
| 1 | **UNIT – I: Introduction**  1.1 Introduction  1.2 Two Orthogonal view of software.  1.3 Object oriented system development methodology.  1.4 why an object oriented  1.5 Overview of the unified approach.  1.6 An object oriented philosophy  1.7 Basic concept of object  1.8 Attributes object state and properties. | Ref.1-11 |
| 2-3 | **2.1 An overview of UML.**  2.2 Where Can the UML Be Used?  2.3 Building Blocks of the UML.  2.5 Relationships in the UML  2.4 Diagrams in the UML.  2.4.1 Use Case Diagrams  2.4.2 Class Diagrams  2.4.3 Sequence diagrams  2.4.4 State chart diagrams  2.4.5Activity diagrams  2.4.6 Component diagram  2.4.7 Deployment diagram  2.5 Diagram extensions | Ref.89-112 |
| 4-6 | **UNIT – III: Requirements Elicitation**  3.1 An overview of requirements elicitation.  3.2 Requirements elicitation concepts  3.2.1 Functional requirements  3.2.2 Nonfunctional and pseudo requirements  3.2.3 Levels of description  3.2.4 Correctness, completeness, consistency, clarity, and realism  3.2.5 Verifiability and traceability  3.3 Requirements elicitation activities.  3.3.1Identifying actors  3.3.2 Identifying scenarios  3.3.3Identifying use cases  3.3.4Refining use cases  3.3.5 Identifying relationships among actors and use cases  3.3.6 Identifying initial analysis objects  3.3.7Identifying nonfunctional requirements  3.4 Managing requirements elicitation  3.4.1Eliciting information from users:  3.4.2 Validating requirements: Usability testing  3.4.3 Documenting requirements elicitation | Ref.**121-151** |
| 7-9 | **4.1 An Overview of Analysis.**  4.2 Analysis Concepts  4.2.1 Entity, Boundary, and Control Objects  4.2.2Association Multiplicity Revisited  4.2.3Qualified Associations  4.2.4Generalization  4.3Analysis Activities: From Use Cases to Objects  4.3.1 Identifying Entity Objects  4.3.2 Identifying Boundary Objects  4.3.3 Identifying Control Objects  4.3.4Modeling Interactions between Objects: Sequence Diagrams  4.3.5 Identifying Associations.  4.3.6 Identifying Attributes  4.3.7Reviewing the Analysis Model | Ref 173-206 |
| 10-12 | **UNIT – V: Object oriented System Design**  5.1 An overview of system design.  5.2 System design concepts.  5.3 System design activities: From objects to subsystems  5.4 Documenting system design  5.5 An overview of object design  5.6 Object design concepts  5.7Object design activities  5.8 Managing object design  5.9 Documenting object design. | Ref. 223-252 |
| 13-14 | **UNIT – VI: Software Quality Assurance**  6 .1an overview of testing  6.2Testing concepts  6.3 Testing activities  6.4 Managing testing  6.5 impact of object oriented testing | Ref. 325-352 |