

COURS OUTLINE

SOFTWARE DESIGN AND MODELLING

Courses Title	Software Design and Modelling (SE 3140)					
Instructor/	Engr. Mangong Clement Fosah					
contact N^o	+237 653 519 879 – mangong.clement@ictuniversity.org					
Duration of Course	Level	Course Code	Credit value	Course type	Semester	Department
	III	SE 3140			Fall	ICT
Course Description	<p>This course begins by placing design and modelling in the context of the various software engineering processes in widespread use today. In the first part, this course teaches students to produce (initially straightforward) designs and to document them using UML Models; both on paper and with an appropriate tool. And in the second part, it focuses on identifying and producing good designs. Finally, model-driven development modelling will be practice and the transformation from one model to another.</p>					
Course Objective	<p>Upon completing this course, students will get significant understanding of software modelling and design domain, straightforward application design, design patterns, and effective object-oriented designs when converting one model to another.</p>					
Expected Outcome	<p>Student will be able to</p> <ul style="list-style-type: none"> ○ Understand the importance of modelling and software design in the software development life cycle ○ Understand the big picture of UML; its history and components ○ Analyse software requirement and produce a several types of models (use case, static, dynamic interaction Models) ○ Produce a straightforward software design (a design from scratch) using the UML models ○ use of UML tools and on paper in designing a software model ○ Relate software code to its model and model back to the code ○ Apply design basic structural, creational and behavioural patterns to a software design ○ Apply design principles to object oriented software designs ○ Apply the principles of transforming from one model to another ○ Implement a model driven development ○ Have a project view about the software engineering future 					

WEEKLY WORK LOAD	TOPICS (Content)	SPECIFIC OBJECTIVES	METHOD OF EVOLUTION	METHOD OF TEACHING
Part I : INTRODUCTION SOFTWARE DESIGN AND MODELLING				
Lesson 1	INTRODUCTION TO SOFTWARE DESIGN AND MODELING <ul style="list-style-type: none"> - Understanding basic modelling terms - The importance of modelling - The need for a model - Impact of software fault - The basics of software Design 	To introduce software design and modelling, its importance, impact of its presence in a software development.	- Oral questioning, discussion and response	- Discussion and lectures (Presentation)
Lesson 2	The Big Picture and History of UML <ul style="list-style-type: none"> - UML usage intention, parts, supporting tools, specification - UML history and standardization - UML diagrams - Use case diagram - Classes and objects - Class diagrams - Interactive diagrams - State machine diagrams 	<ul style="list-style-type: none"> - Understand the big picture of UML; its history and components -To understand the concept of UML; mastering the UML tools and its models/ diagrams. 	Lesson, Exposer, And Assignment	Presentation, discussions
Part II : SOFTWARE MODELING AND DESIGN OF STRAIGHTFORWARD DESIGNS				
Lesson 3	I – Use case modelling <ul style="list-style-type: none"> - Requirement modelling - Use cases - Actors - The Include Relationship - The Extend Relationship - Exercises 	<ul style="list-style-type: none"> - To give an overview of software requirements analysis and specification, describing the use case approach to defining functional requirements, 	- Exposer , exercises, Assignment and class hands-on	Presentati on, exercises, discussions, and hands-on

Lesson 4	II- Static Modelling <ul style="list-style-type: none"> - Association between classes - Static modelling and the UML - Modelling external classes - Static modelling of Entity classes 	<ul style="list-style-type: none"> - To describe the different kinds of associations between classes; describing the whole/part relationships and generalization/specialisation hierarchies. 	<ul style="list-style-type: none"> - Exposer , exercises, Assignment and class hands-on 	Presentati on, exercises, discussions, and hands-on
Lesson 5	III- Object and class structuring <ul style="list-style-type: none"> - Object and class structuring criteria - Modelling application classes - Object and class structuring categories - External classes and software boundary classes - Boundary classes and objects - Entity classes and objects - Application login classes and objects 	<ul style="list-style-type: none"> - To give guidelines on how to determine the objects in the system, the object structuring criteria; objects and classes of a modelling system determination and classification. 	Exposer, exercises, Assignment and class hands-on	Presentati on, exercises, discussions, and hands-on
Lesson 6	IV. Dynamic interaction Modelling <ul style="list-style-type: none"> - Object interaction Modelling - Message sequence Numbering on interaction diagram - Dynamic interaction modelling 	<ul style="list-style-type: none"> - To describe how, for each use case, an interaction diagram is developed to depict the object that participate in the use case and the sequence of messages passed between them. 	<ul style="list-style-type: none"> - Exposer, exercises, Assignment and class hands-on 	Presentati on, exercises, discussions, and hands-on
Lesson 7	V. Finite state Machine <ul style="list-style-type: none"> - finite state machines and state Transitions - Event and Guard conditions - Actions - Hierarchical State charts - Developing a state chart from a use case 	<ul style="list-style-type: none"> - To show the benefits of hierarchical state charts by starting with the simplest form of flat state chart and gradually showing how it can be improved upon to achieve the full modelling power of hierarchical state charts. 	<ul style="list-style-type: none"> - Exposer, exercises, Assignment and class hands-on 	Presentati on, exercises, discussions, and hands-on

Lesson 8	RELATING (JAVA) CODE AND (UML) MODELS	- To transform one UML model to another, and particularly from the source code to the model and verse versa.	- Exposer , exercises, Assignment and class hands-on	Presentation, exercises, discussions, and hands-on
Part III : DESIGN PRINCIPLES AND PATTERNS				
Lesson 9	I. DESIGN PRINCIPLES <ul style="list-style-type: none"> - Why design principle? - Symptoms of rotting design - Causes of design rotting - The SOLID design principles 	- To understand and implement the concept of SOLID to the design of a software	- Exposer , exercises, Assignment and class hands-on	Presentation, exercises, discussions, and hands-on
Lesson 10	II. DESIGN PATTERNS <ul style="list-style-type: none"> - Why patterns? - Why design patterns? - Design patterns Name, the solution and results and consequences of applying the pattern 	- To understand and implement the of several types of design patterns to a straight forward design;	- Exposer , exercises, Assignment and class hands-on	Presentation, exercises, discussions, and hands-on
REVISION, EXPOSERS AND EXAMINATION				

Recommended Textbooks

- *Software Modelling & Design: UML, Use cases, Patterns & Software Architecture ; Hassan Gomaa*
- *Head First Design Patterns : A Brain-Friendly Guide; Eric Freeman & Elisabeth Robson with Kathy Sierra & Bert Bates*

Signature