Comp201 Syllabus

1 Introduction Lecture 1 (week 1)

- 1.1 What is software engineering
- 1.2 Why Software Engineering
- 1.3 Software engineering tasks
- 1.4 Software
- 1.5 Software engineers
- 1.6 Software failure
- 1.7 Good software attributes
- 1.8 Professional and Ethical Responsibility

2 Process Lecture 2 (week 1)

- 2.1 What is a process?
- 2.2 Software process Waterfall, evolutionary development
- 2.3 Agile and SCRUM

3 Software processes Lecture 3 (week 1)

- 3.1 Requirements analysis
- 3.2 Software design
- 3.3 Programming
- 3.4 Testing and QA

4 Software requirements Lecture 4 (week 2)

- 3.1 Functional and non-functional requirements
- 3.2 Specifying non-functional requirements

5 Introduction to coursework 1.1 Lecture 5 (week 02)

- 5.1 Use case analysis

6 Requirements Lecture 6

- 6.1 Hotel booking system case study

7 Requirements engineering process Lectures 7 and 8 (week 3)

- 7.1 Requirements elicitation
- 7.2 Stakeholders/viewpoints
- 7.3 Ethnography
- 7.4 Security requirements

8 System modelling (requirements modelling) Lecture 9 (week 3)

- 8.1 Introduction to modelling
- 8.2 Data, process and architectural models
- 8.3 State machines and state modelling

9 System modelling Lecture 10 (week 4)

- 8.1 Mealy machines, Moore machines and Petri net
- 8.2 Semantic data models, data dictionaries and object models

10 Petri nets Lectures 11 and 12 (week 4)

- 9.1 introduction to petri nets
- 9.2 Petri net case studies
- 9.2 Colour and timing

11 Design methodology Lecture 13 (week 5)

12 Design methodology Lecture 14 (week 5)

13 Design (Distributed systems architecture) Lecture 15 (week 5)

14 Design (Distributed systems architecture) Lecture 16 (week 6)

15 OO design concepts Lecture 17 (week 6)

16 Hotel booking case study) data and class analysis Lecture 18 (week 6)

17 OO design case study Lecture 19 (week 7)

18 Class models Lecture 20 (week 7)

19 Class models continued Lecture 21 (week 7)

20 Interaction diagrams Lecture 22 (week 8)

21 Verification validation Lecture 23 (week 8)

22 Software testing Lecture 24 (week 8)

23 Software testing Lecture 25 (week 9)

24 Project management Lecture 26 (week 9)

25 Software cost estimation Lecture 27 (week 9)

26 Revision and review (week 10)