2024 / 25

School of Science and Computing
+353 (0)51 302037
Eleanor.Reade@setu.ie
www.wit.ie/schools/science_computing

Ollscoil Teicneolaíochta an Oirdheiscirt

South East Technological University

Module Descriptor

Design Patterns (Computing and Mathematics)

Design Patterns (A13563)

Short Title: Design Patterns
Department: Computing and Mathematics
Credits: 10 **Level:** Postgraduate

Description of Module / Aims

This module builds on an undergraduate-level software development knowledge base, the central focus of the module is to broaden the design vocabulary of the student to incorporate best practice in object oriented software development. The module will revisit core design principles and frame these in the context of design patterns. A set of patterns are examined in detail, both in isolation and in the context of integrated applications. Particular attention is paid to relating patterns and assessing design tradeoffs . Alternative Pattern classifications are examined. The role of refactoring in is examined and pattern-based refactoring techniques are presented. Broader software architectural issues are explored and attention is given to relating the scope of architectural styles to the design pattern literature.

Programmes

		stage/semester/status
COMP-0056	MSc in Computer Science (Enterprise Software Systems) (WD_KCESS_R)	1 / 0 / E
COMP-0056	MSc in Computing (Information Systems Processes) (WD_KISYP_R)	1 / 0 / E

Indicative Content

- Review of Design Principles (SOLID)
- Refactoring: Code smells; named refactorings and the role of Test Driven Development (TDD)
- Creational Patterns: Singleton; Factory; Prototype
- Behavioural Patterns: Command; Observer; Strategy; Template Method; State; Iterator; Chain of Responsibility
- Structural Patterns: Facade; Proxy; Bridge; Composite; Adapter
- Pattern Based Refactoring: Refactoring towards and away from specific patterns
- POSA patterns
- Architectural Styles and Architectural viewpoints

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Critique core design principles and assess the quality of a design with respect to these principles.
2. Integrate these core principles in the design of object oriented systems.
3. Evaluate a range of design patterns and comprehend a design presented using this vocabulary.
4. Recommend and intergrate refactoring techniques in the context of design patterns.
5. Appraise the broader scope addressed by Architectural Styles and relate design patterns to these styles.

Learning and Teaching Methods

- This module will be presented by a combination of lectures and computer-based practicals whilst capitalising on a web-enhanced learning environment.
- The lectures will be used to introduce new topics and their related concepts.
- A cooperative learning/peer tutoring (i.e. problem solving / class discussion) approach will be adopted during the sessions.
- Self-directed learning will be encouraged throughout the duration of the module.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	24	24
Practical	24	24
Independent Learning	222	222

Assessment Methods

	Weighting	Outcomes Assessed
Final Written Examination	50%	3,4,5
Continuous Assessment	50%	
Assignment	50%	1,2,3

Assessment Criteria

<40%: Inability to recognise and use any design patterns. Unable to implement applications using design patterns.

40%–59%: Able to recognise and use design patterns to a moderate degree. Able to implement a substantial application consulting a catalogue of design patterns.

60%–69%: Able to recognise, comprehend and apply varying design patterns to a high degree. Able to implement a complex application using a range of design patterns from multiple sources.

70%–100%: All of the above to an excellent level. Demonstrates advanced utilisation of patterns and architectural thinking.

Supplementary Material(s)

- Gamma, E., R. Helm, R. Johnson and J. Vlissides. *Design Patterns*. New York: Addison-Wesley, 1994.
- Holub, A. *Holub on Patterns*. New York: APress, 2005.
- Kerievsky, J. *Refactoring to Patterns*. New York: Addison-Wesley, 2004.
- Martin, R.C. *Agile Software Development, Principles, Patterns, and Practices*. New York: Prentice Hall, 2002.
- Martin, R.C. *The Clean Coder: A Code of Conduct for Professional Programmers*. 1st ed.. New York: Prentice Hall, 2011.
- Martin, R.C. and M. Martin. *Agile Principles, Patterns, and Practices in C#*.. NJ: Pearson Education, 2007.
- Osmani, A. *Learning JavaScript Design Patterns*. New York: O'Reilly, 2015.

Requested Resources

- Computer Lab: BYOD Lab