

Unit 20:

Advanced Programming

| | |
|---------------------|-------------------|
| Unit code | Y/615/1651 |
| Unit level | 5 |
| Credit value | 15 |

Introduction

Features of programming languages that are considered advanced are used to develop software that is efficient; it can affect the performance of an application as well as the readability and extensibility of the code, improving productivity and therefore reducing cost. Many commercial applications available today, whether for productivity or entertainment, will have used one or more design pattern in their development. A design pattern is a description of how to solve a problem that can be used in many different situations and can help deepen the understanding of object-orientated programming and help improve software design and reusability.

The aim of this unit is to familiarise students with these features and their best practices to ensure that their code is in line with industry standards.

Among the topics included in this unit are: object-orientated programming; polymorphism, encapsulation, class aggregation/association, constructors/destructors, inheritance, abstract classes, interfaces, containers, generics, introduction to design patterns and Unified Modelling Language (UML).

On successful completion of this unit students will be able to write code in an object-orientated fashion using design patterns where necessary and be able to model their code structure in UML class diagrams. As a result they will develop skills such as communication literacy, critical thinking, analysis, reasoning and interpretation, which are crucial for gaining employment and developing academic competence.

Learning Outcomes

By the end of this unit students will be able to:

- LO1. Examine the key components related to the object-orientated programming paradigm, analysing design pattern types.
- LO2. Design a series of UML class diagrams.
- LO3. Implement code applying design patterns.
- LO4. Investigate scenarios with respect to design patterns.

Essential Content

LO1 **Examine the key components related to the object-orientated programming paradigm, analysing design pattern types**

Outline the object-orientated paradigm characteristics:

Encapsulation, polymorphism, constructors/destructors, sub objects, abstract/concrete, interface, method redefinition, generics/templates, containers.

Object-orientated class relationships:

Generalisation/inheritance, realisation, dependency, aggregation, composition.

Design patterns:

Creational, structural and behavioural.

LO2 **Design a series of UML class diagrams**

UML class design:

Analyse a code scenario and utilise a suitable UML tool to develop class diagrams.

LO3 **Implement code applying design patterns**

Implementation:

Using an appropriate language & IDE to develop code that implements design patterns and utilises techniques to produce secure code.

LO4 **Investigate scenarios with respect to design patterns**

Review the usage of design patterns:

Relating design patterns to a range of given scenarios

Learning Outcomes and Assessment Criteria

| Pass | | Merit | Distinction |
|--|---|-------|---|
| L01 Examine the key components related to the object-orientated programming paradigm, analysing design pattern types | | | D1 Analyse the relationship between the object-orientated paradigm and design patterns. |
| P1 Examine the characteristics of the object-orientated paradigm as well as the various class relationships. | M1 Determine a design pattern from each of the creational, structural and behavioural pattern types. | | |
| L02 Design a series of UML class diagrams | | | D2 Analyse how class diagrams can be derived from a given code scenario using a UML tool. |
| P2 Design and build class diagrams using a UML tool. | M2 Define class diagrams for specific design patterns using a UML tool. | | |
| L03 Implement code applying design patterns | | | D3 Evaluate the use of design patterns for the given purpose specified in M3. |
| P3 Build an application derived from UML class diagrams. | M3 Develop code that implements a design pattern for a given purpose. | | |
| L04 Investigate scenarios with respect to design patterns | | | D4 Critically evaluate a range of design patterns against the range of given scenarios with justification of your choices. |
| P4 Discuss a range of design patterns with relevant examples of creational, structural and behavioural pattern types. | M4 Reconcile the most appropriate design pattern from a range with a series of given scenarios. | | |

Recommended Resources

Textbooks

Freeman, E. et al. (2008) *Head First Design Patterns*. 4th Ed. United States of America: O'Reilly Media.

Gamma, E. et al. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. 1st Ed. New Jersey: Addison-Wesley.

McLaughlin, B.D. et al. (2007). *Head First Object-Oriented Analysis and Design*. 1st Ed. United States of America: O'Reilly Media.

Links

This unit links to the following related units:

Unit 1: Programming

Unit 19: Data Structures & Algorithms

Unit 28: Prototyping

Unit 41: Analytic Architecture Design