New College of the Humanities

Programming Design Paradigm Course Descriptor

Course Title	Programming Design Paradigm	Faculty	Philosophy
Course code	NCHCS760	Course Leader	Dr Alexandros Koliousis
Credit points	15	Teaching Period	Any
FHEQ level	Level 7	Date approved	September 2020
Compulsory/ Optional	Compulsory		
Pre-requisites	None		
Co-requisites	None		

COURSE SUMMARY

This course provides an intensive tour of programming and design. It exposes students to different ways of thinking of a software problem and designing robust solutions using a mix of programming design paradigms (for instance, by discussing and comparing object-oriented and functional design paradigms). The course concentrates on object-oriented program design and the design of abstractions that support the design of reusable software and libraries.

Students will learn how to explain and defend their design and programming decisions to their peers. In this sense, this course trains students to see programming as a "people's discipline".

COURSE AIMS

The aims of this course are:

- Review typical object-oriented concepts such as information hiding, encapsulation and various forms of polymorphism.
- Contrast the use of inheritance and composition as dual techniques for software reuse.
- Provide a deeper understanding of object-oriented design using the use of graphical design notations such as UML and object-oriented design patterns.
- Emphasise testing, specifically unit testing of components.
- Write clean, well-structured, well-documented code that facilitates reuse, minimises code duplication and make it "future-proof".

LEARNING OUTCOMES

On successful completion of the course, students will be able to:

KNOWLEDGE AND UNDERSTANDING

K1d Master advanced aspects of software design paradigms.

K2d Master object-oriented solutions to solve small and moderately sized practical problems.

K3d Choose, include, and implement design patterns appropriately to design solutions to problems and understand the impact of the design decisions on the technical, social and management dimensions of software.

SUBJECT SPECIFIC SKILLS

S1d Implement a given software design with clean, understandable and well-documented code in the Java programming language, with appropriate unit testing.

S2d Become a sophisticated software developer, familiar with best practices to deliver reusable and extensible code.

S3d Create, refine and express a design in graphical notation such as UML class diagrams.

TRANSFERABLE AND PROFESSIONAL SKILLS

T1d Critically review and defend the design choices made in existing software libraries and frameworks to a group of peers, identify limitations and propose ways for improvement.

T2d Generate appropriate documentation for developed solutions.

T2d Consistently apply an excellent level of technical proficiency in written English, using an advanced application of scholarly terminology, that demonstrates the ability to deal with complex issues both systematically and with sophistication.

T3d Lead or participate in the design and implementation of software artefacts.

TEACHING AND LEARNING

Teaching and learning strategies for this course will include:

- 30 hours of full-cohort lectures
- 20 hours of lab-based tutorials
- 1 office hour per teaching week

There will be three 1-hour lectures per teaching week. Two 1-hour lab sessions will give students the opportunity to work on their assignments with the help of the course leader and teaching assistants.

Course information and supplementary materials are available on the College's Virtual Learning Environment (VLE).

Students will also attend the formal meeting, Collections, in which they will receive constructive and developmental feedback on their performance.

Students are required to attend and participate in all the formal and timetabled sessions for this course. Students are also expected to manage their directed learning and independent study in support of the course.

EMPLOYABILITY SKILLS

- Communication Skills
- Programming skills

ASSESSMENT

FORMATIVE

Students will be formatively assessed during the course by means of set assignments. These do not count towards the end of year results but will provide students with developmental feedback. Set assignments will also amplify problem-solving skills and develop software components that form part of the coding assignments.

SUMMATIVE

The assessment will consist of two written coding assignments, which the student will have to do to the set guidelines for coding.

AE:	Assessment Activity	Weighting (%)	Online submission	Coding	Length
1	Coding assignment	50	No	Yes	Code and 2500-word explanation
2	Coding assignment	50	No	Yes	Code and 2500-word explanation

The coding assignments will be assessed in accordance with the assessment aims set out in the Programme Specification.

FEEDBACK

Students will receive formal feedback in a variety of ways: written (including via email correspondence); oral (within one-to-one tutorials or on an *ad hoc* basis) and indirectly through discussion during group tutorials.

Feedback is provided on summative assessment and is made available to the student either via email, the VLE or another appropriate method.

INDICATIVE READING

Note: Comprehensive and current reading lists for courses are produced annually in the Course Syllabus or other documentation provided to students; the indicative reading list provided below is used as part of the approval/modification process only.

BOOKS

Joshua Bloch. 2017. Effective Java (3rd. ed.). Addison-Wesley Professional, USA

Walter Savitch. 2016. Absolute Java 6th edition (6th. ed.). Pearson, USA

ELECTRONIC RESOURCES

The Java Tutorials: Learning the Java language, by Oracle. Last accessed: August 2020

INDICATIVE TOPICS

Students will study the following topics:

- Object-oriented program design with classes
- Class diagrams
- Testing with Junit
- Handling exceptions
- Primitive versus reference types
- Information hiding
- Polymorphism, encapsulation and inheritance in object-oriented programming
- Lists and iterator patterns

Title: NCHCS760 Programming Design Paradigm

Approved by: Academic Board

Location: Academic Handbook/Programme specifications and Handbooks/ Postgraduate Programme Specifications/MSc Computer Science Programme Specification/Course Descriptors

Version number	Date approved	Date published	Owner	Proposed next review date	Modification (As per AQF4) & category number
2.0	January 2022	April 2022	Dr Alexandros Koliousis	April 2025	Category 3: Changes to Course Learning Outcomes
1.0	September 2020	September 2020	Dr Alexandros Koliousis	April 2025	