MODULE SYLLABUS

PROGRAMMING PRINCIPLES AND ALGORITHMS

MODULE CODE	CCS1110
MODULE TITLE	Programming Principles and Algorithms
PROGRAMME	BSc (Hons) in Computer Science
DEPARTMENT	Computer Science
CREDITS	10
STAGE OF STUDY	1
SEMESTER/SESSION	Fall 2021-2022
RE-ASSESSABLE	YES
COMPENSATABLE	YES
LOCATION	Thessaloniki
STAFF	Dr Ioanna Stamatopoulou
E-MAIL	istamatopoulou@york.citycollege.eu
STAFF OFFICE	L. Sofou bldg., 6th floor
ACCREDITATION	The programme is accredited by: the British Computer Society (BCS)

DESCRIPTION

This module is an introduction to computer programming covering the fundamental constructs and practices required for the procedural development of software, using the Java programming language as a vehicle. Emphasis is placed on problem analysis and developing algorithmic thinking skills.

AIMS

This unit aims to:

A1 teach students how to write, compile, and execute Java programs; and

A2 fundamental programming concepts and techniques.

LEARNING OUTCOMES

By the end of the unit, a student will be able to:	Link to aims
LO1 identify syntactical and logical errors in Java code;	A1
LO2 determine, define, and use variables of various primitive types;	A2
LO3 make use of standard Java library classes' methods;	A2
LO4 identify cases requiring the use of control and repetition structures; and	A2
LO5 apply algorithmic techniques to solve simple problems.	A1, A2

HOW DOES THIS MODULE FIT INTO THE CURRICULUM?

This module sets the foundation for algorithmic thinking and problem-solving, and procedural programming, which is necessary for any computer scientist. The principles learned are applicable to any other high-level programming language students may learn in the future and are will also be used when students will be introduced to Object-Oriented Programming in the next semester. Finally, Java, as the actual language learnt, will be useful for a number of future modules, such as Data Structures and Algorithms in Stage 2.

TEACHING & LEARNING METHODS

Total Contact Hours: 20

The following teaching & learning methods will be employed:

The unit is delivered through two 2-hour sessions every week.

- During the first 2 hours the theoretical concepts are taught and students have the opportunity to immediately work on small example/abstract exercises so that they practice the Java syntax.
- During the second 2 hours, where students are split into two smaller groups, they work entirely on lab exercises that build up in complexity, so that they develop their algorithmic thinking and problem solving skills.

ASSESSMENT METHODS

#	Students will be assessed by:	Submission Week	% contribution	LOs assessed
1	Assessed Lab: open-access, in-class	CW6	35%	LO1–LO3
2	Assessed Lab: open-access	RA15	65%	LO1–LO5

FEEDBACK PROVISION

The following methods will be used to provide feedback to students:

- Formative feedback: Students have the opportunity to receive formative (group as well as individual) feedback every week during the labs as they get to work alone or in pairs on given exercises, before we solve them together.
- Summative feedback will be provided to the entire class after the submission of all pieces of coursework: all exercises will be solved so that students see the “model answers” that are expected and compare them with what they did.

The feedback handbook found at https://goo.gl/Zy2roA aims to give you a better understanding of feedback; what it is for and how to use it.

ACCESS TO MODULE MATERIAL (Notes, handouts, announcements, etc.)

All material used in this module's classes are available in electronic form through Google Classroom with class code **xlgpalb**

RECOMMENDED TEXTBOOK(S)

Paul Deitel, Harvey Deitel, *Java How To Program (late objects)*, 10th edition, Prentice Hall, 2015.

LIST OF REFERENCES / ADDITIONAL RECOMMENDED READING

Brian P. Hogan, Exercises for programmers: 57 challenges to develop your coding skills, The Pragmatic Bookshelf, Dallas, Texas, 2015.

OUTLINE

WEEK	
#1	**Writing a Java program** Compiling, Executing **Output** to the user
#2	**Variables** Declaration, initialization, value assignment Primitive Data types **Arithmetic operators** User **Input**
#3	**No classes**
#4	**Control Structures** if…then…else and switch statements Equality and comparison operators
#5 - #6	**Repetition Structures** for, while, and do…while

EMPLOYABILITY PROFILE

This module contributes to your employability profile by enhancing the following Graduate Attributes:

Knowledgeable

Information & Digitally Literate