MODULE TITLE	**Programming**	**CREDIT VALUE**	**15**
MODULE CODE	**ECM1400**	**MODULE CONVENER**	**Dr Matt Collison (Coordinator)**

DURATION: TERM	**1**	**2**	**3**
DURATION: WEEKS	**11**	**0**	**0**

Number of Students Taking Module (anticipated)	**97**

DESCRIPTION - summary of the module content

We use computers in almost all aspects of our daily lives and throughout science, so it is easy to take them for granted. However, in order that we can use computers to solve new problems and create new things, we have to be able to program them. This module introduces you to programming and problem solving with a computer. You will learn how to formulate an algorithm to solve a problem, and you will acquire the skills to write, test and debug programs.

AIMS - intentions of the module

This module is an introductory course in computer programming and will introduce you to the fundamental concepts of computer algorithms and programming, with a strong emphasis on practical implementation. You will also learn how to apply analytical and problem-solving skills to the design and implementation of small applications.

INTENDED LEARNING OUTCOMES (ILOs) (see assessment section below for how ILOs will be assessed)

On successful completion of this module, **you should be able to**:

Module Specific Skills and Knowledge:

1 design an algorithm, using sequence, iteration and selection;

2 write, compile, test, and debug a computer program;

3 explain how a program written in a procedural language is translated into a form that allows it to be executed on a computer;

4 systematically test your programs;

5 document software to accepted standards;

6 design an algorithm, using a divide and conquer strategy;

7 demonstrate familiarity with basic numerical and discrete algorithms;

Discipline Specific Skills and Knowledge:

9 systematically break down a problem into its components;

10 understand and choose appropriate programming techniques.

Personal and Key Transferable/ Employment Skills and Knowledge:

11 analyse a problem and synthesise a solution;

12 use technical manuals and books to interpret specifications and technical errors.

SYLLABUS PLAN - summary of the structure and academic content of the module

Problem solving and programming overview: algorithms, flow charts; pseudo-code; compilers and interpreters;

Python as a language: statements, comments and simple arithmetic operations;

variables, and data types;

functions: encapsulation and abstraction; arguments and return values; namespaces and scope;

sequences and iteration: lists, loops, nested loops; accumulation as a programming idiom;

flow control: conditional expressions, while loops; searching by bisection and bracketing;

integer and floating point representation; numerical precision;

mutable and immutable variables, and sequences: tuples, lists and strings;

simulation: pseudo-random numbers;
larger programs; encapsulation and program organisation;

input/output and exceptions;

recursion: divide and conquer algorithms; memoisation;

associative arrays, hashing and dictionaries;

basic functional programming constructions: map, filter, reduce, anonymous functions;

searching and sorting: linear search versus bisection; insertion sort, bubble sort, merge sort.

LEARNING AND TEACHING

LEARNING ACTIVITIES AND TEACHING METHODS (given in hours of study time)

Scheduled Learning & Teaching Activities	42.00	**Guided Independent Study**	108.00	**Placement / Study Abroad**	0.00

DETAILS OF LEARNING ACTIVITIES AND TEACHING METHODS

Category	Hours of study time	Description
Scheduled learning and teaching activities	22	Lectures
Scheduled learning and teaching activities	20	Workshops/tutorials
Guided independent study	66	Individual assessed work
Guided independent study	42	Lecture and assessment preparation

ASSESSMENT

FORMATIVE ASSESSMENT - for feedback and development purposes; does not count towards module grade

Form of Assessment	Size of Assessment (e.g. duration/length)	ILOs Assessed	Feedback Method
Workshop Exercises	2 hours per week	1-12	Model Answers and verbal feedback

SUMMATIVE ASSESSMENT (% of credit)

Coursework	100	Written Exams	0	Practical Exams	0

DETAILS OF SUMMATIVE ASSESSMENT

Form of Assessment	% of Credit	Size of Assessment (e.g. duration/length)	ILOs Assessed	Feedback Method
1 Continuous Assessment	100	80 hours	All	Written

DETAILS OF RE-ASSESSMENT (where required by referral or deferral)

Original Form of Assessment	Form of Re-assessment	ILOs Re-assessed	Time Scale for Re-assessment
Coursework	Coursework (100%)	All	Completed over the summer with a deadline in August

RE-ASSESSMENT NOTES

Reassessment will be by coursework in the failed or deferred element only. For referred candidates, the module mark will be capped at 40%. For deferred candidates, the module mark will be uncapped.

RESOURCES

INDICATIVE LEARNING RESOURCES - The following list is offered as an indication of the type & level of information that you are expected to consult. Further guidance will be provided by the Module Convener

Web based and Electronic Resources:
ELE: http://vle.exeter.ac.uk
Think Python website: http://www.greenteapress.com/thinkpython/
Python language website: http://www.python.org

Reading list for this module:

Type	Author	Title	Edition	Publisher	Year	ISBN	Search
Set	Allen B. Downey	Think Python: How to think like a Computer Scientist		O`Reilly Media	2012	978-1449330729	[Library]
Set	Zelle, J.	Python Programming: An Introduction to Computer Science	2nd Edition	Franklin, Beedle & Associates	2010	978-1590282410	[Library]
Set	Summerfield Mark	Programming in Python3	2nd Edition	Addison Wesley	2010	978-0321680563	[Library]
Set	Hunt, Andrew; Thomas, David	The Pragmatic Programmer	1st	Pearson	1999	978-0201616224	[Library]

CREDIT VALUE	15	ECTS VALUE	7.5

PRE-REQUISITE MODULES	None
CO-REQUISITE MODULES	None

NQF LEVEL (FHEQ)	4	AVAILABLE AS DISTANCE LEARNING	No
ORIGIN DATE	Tuesday 10 July 2018	LAST REVISION DATE	Monday 11 April 2022

KEY WORDS SEARCH	Computer; programming; algorithms; problem solving; Python.