



Intensive Foundations of Computer Science and Programming I Course Descriptor

Course Title	Intensive Foundations of Computer Science and Programming I	Faculty	EDGE Innovation Unit (London)
Course code	NCHNAP445	Course Leader	Professor Scott Wildman (interim)
Credit points	15	Teaching Period	This course will typically be delivered over a 6-week period.
FHEQ level	4	Date approved	June 2020
Compulsory/Optional	Compulsory		
Prerequisites	None		

COURSE SUMMARY

This course introduces the fundamental ideas of computing and programming principles. The course discusses a systematic approach to word problems, including analytic reading, synthesis, goal setting, planning, plan execution, and testing. It presents several models of computing, beginning with functional program design. Learners will explore the Python programming language, its syntax, mathematical functionality and suitability for data analysis applications.

COURSE AIMS

- Train learners in the fundamentals of computing and programming principles.
- Train learners in Python programming.
- Give learners the tools to design and implement basic Python programmes.

LEARNING OUTCOMES

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

KNOWLEDGE AND UNDERSTANDING

- K1a Understand the underlying basic concepts and principles associated with programming languages.
- K2a Understand the basic syntax and structure of a Python programme.
- K3a Use Python file input/output functions to work with directories and files.

SUBJECT SPECIFIC SKILLS

- S1a Write, test and correct basic programs that others can read, understand and modify.
- S2a Break large problems into an appropriate design for implementation.
- S3a Select appropriate data types to represent information.

TRANSFERABLE AND PROFESSIONAL SKILLS

- T1ai Test, evaluate and identify errors in coding.
- T1aii Display a developing technical proficiency of written English skills that demonstrates an ability to communicate clearly and accurately when producing structured and coherent pieces of text.
- T2a Appreciate the impact of data structure and algorithm choice on the running time and storage space needed to run a programme.
- T3a Understand professional and ethical issues and guidelines.

TEACHING AND LEARNING

This is an e-learning course, taught throughout the year.

This course can be offered as a standalone short course.

Teaching and learning strategies for this course will include:

- On-line learning
- On-line discussion groups
- On-line assessment

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

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

The course learning and teaching hours will be structured as follows:

- Off-the-job learning and teaching (6 days x 7 hours) = 42 hours
- On-the-job learning (12 days x 7 hours) = 84 hours (e.g. 2 days per week for 6 weeks)
- Private study (4 hours per week) = 24 hours

Total = 150 hours

Workplace assignments (see below) will be completed as part of on-the-job learning.

ASSESSMENT

FORMATIVE

Learners will be formatively assessed during the course by means of set assignments. These will not count towards the final degree but will provide learners with developmental feedback.

SUMMATIVE

Assessment will be in two forms:

AE	Assessment Type	Weighting	Online submission	Duration	Length
1	Set programming exercises	50%	Yes	Requiring on average 15-25 hours to complete	N/A
2	Practical skills assessment (workplace dataset)	50%	Yes	Requiring on average 15-25 hours to complete	N/A

FEEDBACK

Learners will receive formal feedback in a variety of ways: written (via email or VLE correspondence) and indirectly through online discussion groups. Learners will also attend a formal meeting with their Academic Mentor (and for apprentices, including their Line Manager). These bi- or tri-partite reviews will monitor and evaluate the learner's progress.

Feedback is provided on all summatively assessed assignments and through generic internal examiners' reports, both of which are posted on the VLE.

INDICATIVE READING

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

BOOKS

- Summerfield, M. (2009), *Programming in Python 3: A Complete Introduction to the Python Language*, Upper Saddle River, NJ: Addison-Wesley
- Lutz, M. (2011), *Programming Python*, Beijing; Farnham: O'Reilly
- Allen, B. (2015), *Think Python: How to Think Like a Computer Scientist*. Farnham: O'Reilly

JOURNALS

Learners are encouraged to read material from relevant journals on Computer Systems and Programming as directed by their trainer.

ELECTRONIC RESOURCES

Learners are encouraged to consult relevant websites on Computer Systems and Programming.

INDICATIVE TOPICS

- Basic syntax and semantics of Python
 - Variables and primitive data types
 - Sequential and binary search algorithms
 - Stacks and Queues
-

Title: NCHNAP445 Intensive Foundations of Computer Science and Programming I					
Approved by: Academic Board					
Location: Academic Handbook/Programme specifications and Handbooks/ Undergraduate Apprenticeship Programmes/BSc (Hons) Digital & Technology Solutions Programme Specification/Course Descriptors					
Version number	Date approved	Date published	Owner	Proposed next review date	Modification (As per AQF4) & category number
2.1	May 2022	May 2022	Scott Wildman	June 2025	Category 1: Corrections/clarifications to documents which do not change approved content.
2.0	January 2022	April 2022	Scott Wildman	June 2025	Category 3: Changes to Learning Outcomes
1.0	June 2020	June 2020	Scott Wildman	June 2025	