

Assignment Specification 2025-2026

Module Title	Foundations of Programming and Software Engineering	
Module Code	CSC-44102	
Assessment Type	Assignment	
Assessment Title	Assessment 1	
Weighting (% of module mark)	80%	
Assessment Length (word count or equivalent)	Equivalent of 4000 words	
Submission Deadline (date and time)	14 November 2025	1 pm
Format of Submission	ZIP file	
Type of GenAI Use Permitted in Assessment	Gen-AI-Assisted Reviewer	
Feedback Release Date	04 December 2025	
Staff contact details	Dr Alastair Channon a.d.channon@keele.ac.uk	

Assessment Details:

The Keele University Library Map on the final page of this brief includes a list of classmarks for each subject, and shows the location of each classmark. For the purpose of this assessment, assume that all classmarks are either one or two letters long, and that a range such as “G to GF” consists of (for this example) G, GA, GB, GC, GD, GE and GF. Consider there to be six locations containing books: Ground Floor, Middle Floor, Top Floor Back Left, Top Floor Back Right, Top Floor Front Left and Top Floor Front Right.

First develop a text-based application (without a GUI) that 1. prompts the user (with clear instructions displayed) to select one of three options: whether to enter a subject name or part-name (e.g. “English”), a classmark, or a location; 2. prompts (according to the previous response) the user to enter the subject name/part-name, classmark or location (with location options displayed); and 3. outputs each matching classmark together with its location and subject name(s).

Your program should read in two comma-separated values (CSV) files that you write. The first file should contain the list of subjects and classmarks. The second file should contain a list of classmarks and locations; you can design this file either to contain one classmark per row or a range of classmarks per row. Your program should be able to operate correctly if

subject names, classmarks or locations are changed within, added to or removed from these files.

Develop a second application (for the same task) that uses a TKinter-based graphical user interface (GUI) for both input and output, rather than the text-based console. Minimise the amount of repeated code by separating code common to both part A and part B from user-interface-specific code.

Create a demo video, up to two minutes long, that demonstrates that your text-based and GUI applications function as specified above. Your video should be in a standard format that can be played on the lab PCs.

Submit a single .zip file, containing your Python .py files, CSV files and demo video file, to the KLE drop-box for this assessment.

Module Learning Outcomes:

In this assessment the following module learning outcomes will be assessed:

ILO1: Implement common algorithms to solve programming problems using programming constructs and data structures.

ILO2: Design and implement practical applications using Python to meet a set of requirements.

ILO3: Demonstrate best practices and conventions for writing clean and efficient code.

Assessment Criteria:

Text-based application functions correctly (20%)

GUI application functions correctly (20%)

Video demonstrates that the applications function correctly (20%)

Code quality, including error/exception handling and comments (20%)

Overall design* of the applications (20%)

* Marks here will be awarded for accordance with the design principles established within the course: in particular, the quality of partitioning and the design goals of flexibility, extensibility, maintainability, performance and usability. Note that this need not necessarily imply an object-oriented design. Appropriate levels of abstraction, partitioning, decoupling, and maximisation of functional cohesion, should be the driving forces.

Within each of the above categories, marking will be according to the University's Generic Assessment Criteria for Level 7.

Feedback to Students:

Feedback will be provided via the KLE.

Inclusive Practice:

You have flexibility in producing your work using a development environment of your choice. However, your submission must use the file formats specified.

Use of Artificial Intelligence (AI):

All students can use assistive AI tools to check spelling, grammar and punctuation (i.e., use of in-built spell checkers) in all written assessments excluding examinations and class tests, unless advised otherwise.

For this assessment, you may use GenAI to act as a reviewer to give academic feedback on a draft of your assessment (e.g. identifying areas for improvement). You are not permitted to use GenAI to generate new content. At the top of each code file, you must acknowledge any use of GenAI. For example: "I acknowledge the use of Microsoft Copilot (version GPT-4, Microsoft, <https://copilot.microsoft.com/>) to review a draft of my work and provide feedback".

Academic Misconduct:

Academic misconduct is doing something that could give you an unfair advantage in an assessment. It includes, but is not limited to, the following: plagiarism; collusion; contract cheating; cheating in an examination; falsification of data or sources; falsification of official documents or signatures. The University treats academic misconduct very seriously and penalties will be given for proven cases, including termination of studies in serious cases. It is therefore very important that you understand how to prepare and take assessments honestly. In order to assist you with this there are various resources and help available both as part of your programme of study and also centrally. For more information please visit:

<https://www.keele.ac.uk/students/academiclife/appeals-complaints-conduct/studentacademicconduct/>

Academic Skills Support:

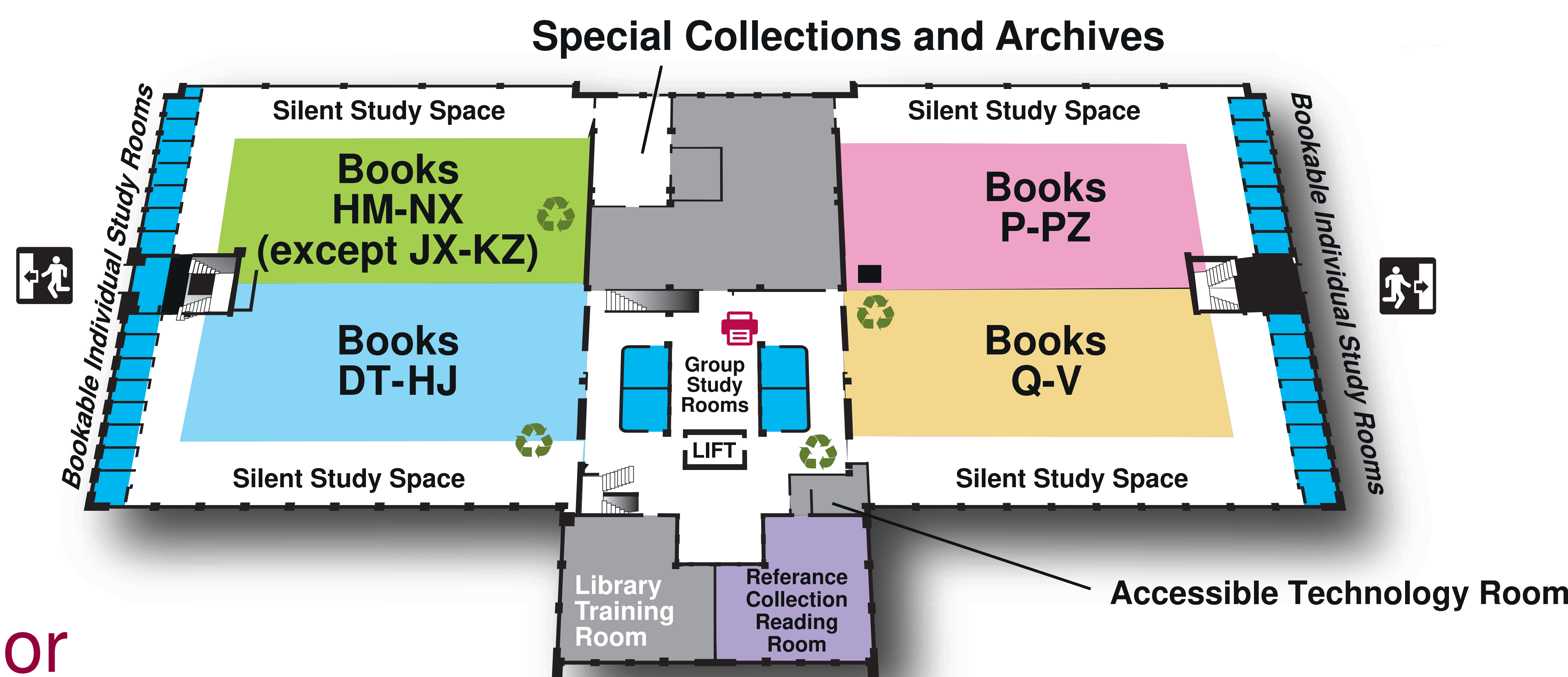
The Academic and Digital Skills team provide a range of additional online resources (e.g., study guides, Sways, Podcasts, workshops etc) to help you with your academic work and assessments. You can find more information [here](#).

Additional information:

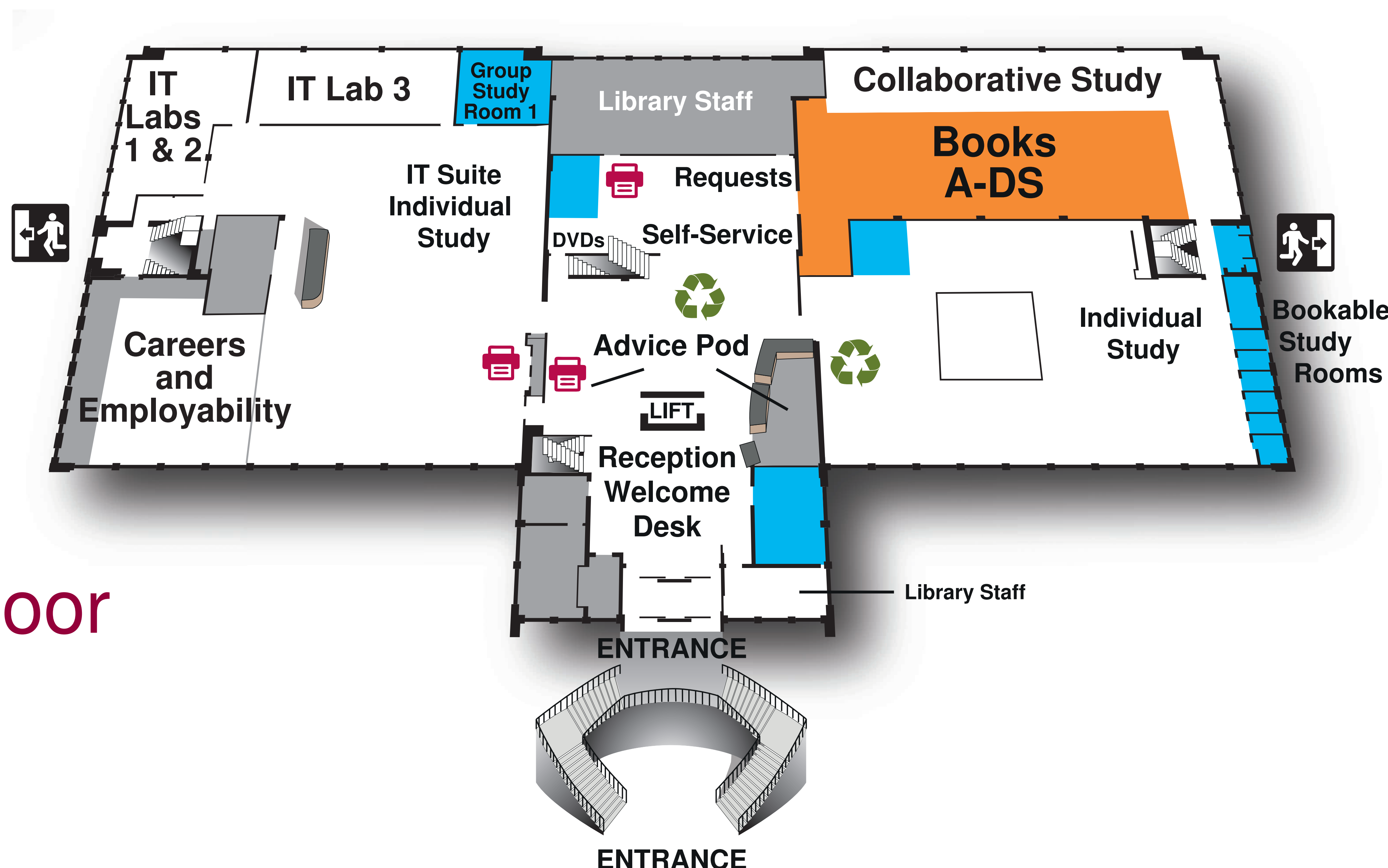
You must complete your work for this assessment without help or feedback from others, including demonstrators. All work that you submit must be your own individual work, and any other person's work or ideas must be appropriately acknowledged. You must be careful not to share your work for this assessment, so it must not appear in a public repository on GitHub or elsewhere.

You can however get help with non-assessment material by making use of the practical sessions timetabled in weeks 5 and 6 (with demonstrators in them) and the weekly virtual help desk sessions. For example, if you are stuck on something to do with the assessment, you might find and try a practical exercise that does a similar thing (e.g. read a CSV file) and get help with that if you need it.

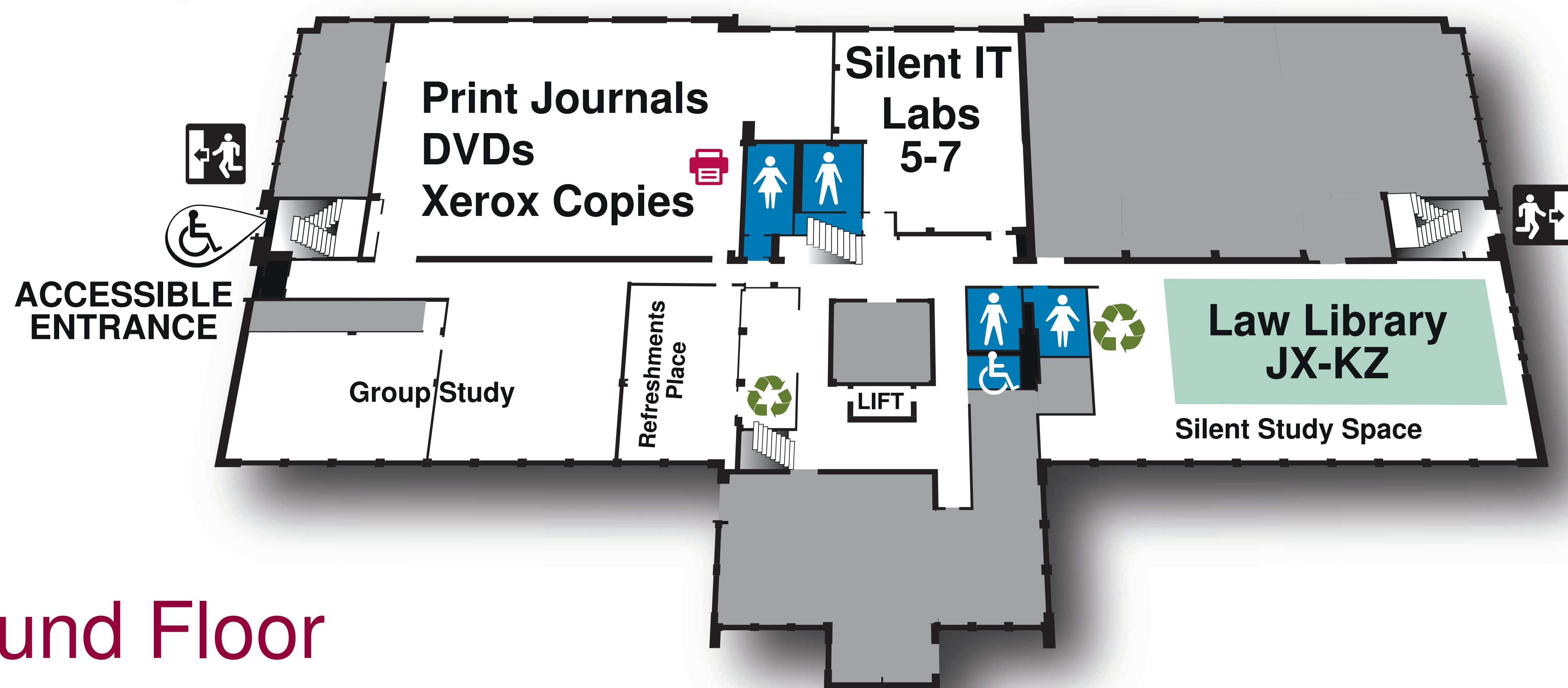
Library Map



Top Floor



Middle Floor



Ground Floor

Toilets Recycling point Printer

Subject Classmark References

Accountancy	HF
American History	E and F
American Literature	PS
Archaeology	CC
Art	N
Astrophysics	QB
Biochemistry	QP and RB
Biology	QH
Biomedical Science	QR and RB
Botany	QK and S
British History	DA
Business	HD to HF
Chemistry	QD
Chinese	PL
Computing	QA and TK
Counselling and Psychotherapy	RC
Criminology	HV
Ecology	QH
Economics	HA to HJ
Education	L
English Language	PE
English Literature	PR
Environmental Studies	GE to GF
Film Studies	PN
Finance	HG to HJ
Forensic Science	HV
French	PC
Geography	G to GF
Geology and Geoscience	QE
German	PF
History	D
Human Biology	QM to QR
Human Resource Management	HD to HF
International Law	JX
International Relations	JZ
Italian	PC
Japanese	PL
Law	K
Management and Marketing	HD to HF
Mathematics	QA
Media Studies	P
Medicinal Chemistry	QP
Medicine	R
Mental Health	RA to RC
Music Technology	M T
Music Technology	ML
Neuroscience	RC
Nursing	RI
Pharmacology	RM
Pharmacy	RS
Philosophy	B to BJ
Physics	QC
Physiology	QP
Physiotherapy	RM
Politics	J to JX
Psychology	BF
Radiography	RC
Religion	BL to BX
Russian	PG
Social Work	HV
Sociology	HM to HT
Spanish	PC
Veterinary Medicine	SF