

# Principles of Programming

**CT4029**

## Module Guide

2023 – 2024

**Zainab Loukil and Nasreen Anjum**

School of Business+

**University of Gloucestershire 2023**

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, including – but not limited to – photocopy, recording, or any information storage and retrieval system, without the specific prior written permission of University of Gloucestershire.

# Table of Contents

Module Information 2

[Learning outcomes 3](#_bookmark0)

[Module Evaluation 3](#_bookmark1)

[Feedback on work 3](#_bookmark2)

Provisional Scheme of work 4

[Assessment : Individual Assignment 7](#_TOC_250000)

Assessment : Requirements 9

[Assessment : Marking criteria 11](#_bookmark3)

Large print copies of this booklet can be provided on request. We can also arrange production of braille or audio versions.

Please email [disability@glos.ac.uk](mailto:disability@glos.ac.uk) if you have any special requirements that you would like to discuss.

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | | | |
| This module studies the fundamental concepts and building blocks of creating, designing and evolving algorithms and translating algorithms into software using a programming language. | | | |
|  | | | |
| **Tutoring Team** | | | |
| *Zainab Loukil*  *Nasreen Anjum* | FHM Computer Lab | | [zloukil2@glos.ac.uk](mailto:zloukil2@glos.ac.uk)  nanjum@glos.ac.uk |
|  | | | |
| **Level** | | 4 | |
| **Credit Value** | | 15 CAT points | |
| **External Examiner** | | TBA | |
|  | | | |
| **Teaching & Learning Approach** | | | |
| **Session** | | Seminar, 3 hours  Generally, 1-hour lecture & 2-hour lab work Lecture, practical exercises, worksheets | |
| **Independent study** | | At least 7 hours per week | |
|  | |  | |

## Learning outcomes

A student passing this module should be able to:

1. Develop an understanding of data structures and programming techniques in context of a programming language.
2. Experience hands-on development of working software
3. Apply problem solving tools and techniques to software projects.
4. Demonstrate skills for software development, documentation, testing & debugging.

## Module Evaluation

### EVALUATION FOR 2022/23

The feedback from last year suggested more guided lab exercises in the form of tutorials should be made regular part of sessions.

#### THE RESPONSE FOR THE CURRENT YEAR IS:

More hands on and tutorial-based exercises will be included in the schedule.

### EVALUATION FOR THE CURRENT YEAR

In this current academic year 2023/24 you will be given the opportunity to undertake a mid-module evaluation which will feed into the course board of studies meeting and will inform module design for the following year. In addition, there will be an independent end of year level evaluation distributed by the University.

## Feedback on work

All students are entitled to both formative and summative feedback during the module. Formative feedback is designed to enable you to improve your work/performance and takes many forms, formative feedback is only applicable for your class exercises not your assessments. Students will have several opportunities to receive feedback on their progress, e.g. during lab sessions, tutorials or seminars; as part of general feedback in lectures; and where appropriate, one-to-one sessions with their Personal Tutors. Summative feedback is received when your submitted coursework has been marked. As a minimum, this will comprise constructive comments from the tutor and a grade.

**Scheme of work**

|  |  |  |
| --- | --- | --- |
| **Semester 1** | | |
| **Session** | **Session Topic** | **Practical work** |
| **1** | Module Overview, Computer Architecture, Introduction to Programming principles | Problem  Solving Strategies, Flowchart, Pseudocode |
| **2** | Basics of Python, Variables and Data Types, Programming Concepts – Control  Structures | Multiple Problem Solving, Loops, Switches, Conditions |
| **3** | Programming Concepts – Passing Values | Functions and Parameters |
| **4** | Design a User Interface Using Python | Graphical User Interface design, Interactive Software implementation using Tkinter |
| **5** | Databases in Python | SQLite Database, Link GUI to SQLite database |
| **6** | Data Structures | Lists and Tuples |
| **7** | File & I/O | Reading/Writing to files |
| **8** | Data Analysis | Python Libraries for Data Science |
| **9** | Arrays and Data visualization | NumPy Arrays, Graphs & Charts using Matplotlib |
| **10** | QR code and Password check in Python | QR code generation and scanning, Password strength check strategies |
| **11** | Recap, Q&A, Extra Practical | |
| **12** | **Assignment Workshop** | |

**Assessment: individual assignment**

|  |  |
| --- | --- |
| **Module code** | **CT4029** |
| **Module title** | **Principles of Programming** |
| **Module tutor** | **Zainab Loukil**  **Nasreen Anjum** |
| **Tutor with**  **Responsibility for this Assessment** | **Zainab Loukil** |
| **Assignment** | **001: 100% Coursework: Individual, standard written (2000 words or equivalent).**  **You will be penalised according to the Academic Regulations for Taught Provision if you exceed the size limit.** |
| **Submission deadline**  **Your attention is drawn to the penalties for late submission; see Undergraduate Modular**  **Handbook.** | **Tuesday 14th November 2023, at 15:00**  **Your attention is drawn to the penalties for late submission; see Academic Regulations for Taught Provision.** |
| **Arrangements for submission** | **Moodle** |
| **Date and location for return of work** | **Written feedback and provisional mark should be within 20 working days of submission.** |
| **Students with Disabilities** | **Alternative assessment arrangements may be made, where appropriate, for students with disabilities. However, these will only be implemented upon the advice of the Disability Advisor. Students wishing to be considered for alternative assessment arrangements must give notification of the disability (with evidence) to the Disability Advisor by the**  **published deadlines.** |

|  |  |
| --- | --- |
| **University Regulations for Assessment** | **All assessments are subject to the Academic Regulations for Taught Provision. These include regulations relating to Errors of Attribution and Assessment Offences. In exercising their judgement, Examiners may penalise any work where the standard of English, numeracy or presentation adversely affects the quality of the work, or where the work submitted exceeds the published size or time limits, or where the work fails to follow normal academic conventions for acknowledging**  **sources.** |
| Assignment Requirements | The purpose of this exercise is to provide the experience of creating interactive software by using Python.  The assessment contributes to all learning outcomes as indicated and will comprise of the following components:  1. Write a report including the design and specification of your project, highlighting the design pattern used for development and its motivation. **(40 marks)**  2. Develop the product (software) using appropriate programming concepts and conventions. **(40 marks)**  3. Develop and document a testing process. **(10 marks)**  4. Prepare a 5 mins video to demonstrate a working prototype of your software with all the features you have developed. **(10 marks)**  **Senario:** Secure Marketing Application with Graphical User Interface using SQLite and QR Code  **Overview:** As a programmer, you have been tasked with developing a secure marketing application with a graphical user interface (GUI) using SQLite database and QR code. The system should allow users to view marketing promotions, scan QR codes to access additional information, and track their interactions with promotions.  **Software Requirements:**   1. Graphical User Interface: You will need to develop a GUI for the marketing application using Python programming language. The GUI should be user-friendly and visually appealing, with at least the following features to include search bar, promotion categories, and a QR code scanner. **(12 marks)** 2. User Authentication: The system should have a secure user authentication process, requiring users to provide their login credentials to access their personal information and view promotions. Passwords must be hashed and stored securely in the SQLite database. **(5 marks)** 3. QR Code Generator and Scanner: The system should be able to generate and scan QR codes and retrieve additional information about promotions, such as product details or discounts. You will need to implement a QR code generator and scanner in the application. QR codes generated should be saved in the local directory of your developed software (application). **(8 marks)** 4. SQLite Database: You will need to create a SQLite database to store promotion and user information. The database should have at least three tables: one for promotions, one for user registration, and another for storing user interactions with promotions. The database should be linked with developed GUI. **(12 marks)** 5. SQL Injection Prevention: The system should be protected against SQL injection attacks. You must ensure that all user inputs are sanitized and validated before being passed to the SQLite database. For this application, any user’s age under 18 is considered as an attack and details should be rejected and not saved in the database. **(3 marks)**   You can use one of the following Python GUI libraries:  · [PyQt5](https://pypi.org/project/PyQt5/)  · [Python Tkinter](https://docs.python.org/3/library/tkinter.html)  · [PySide2](https://pypi.org/project/PySide2/)  · [Kivy](https://kivy.org/doc/stable/guide/basic.html)  It is very important to focus on the consistency of your application. Your Secure Marketing Application can contain one or more windows linked together.  **Report Structure:**  You are asked to write a report with at least the following sections:  **Table of contents**  **1. Introduction (5 marks)**  A short, informal description of the software, its objectives and an indication of the extent to which it has been implemented and any problems faced in its development  **2. Design of system (25 marks)**  Discussion and reflection on the design and development of the software, showing the structure of the software and identifying appropriate features.  Discussion of implementation logic, code snippets, diagrams, and screenshots where needed.  A flowchart of the overall system and ER diagram of the database are required.  **3. Testing the system (10 marks)**  A set of test results to be submitted.  A discussion on failed tests and how much of your code you think your tests cover. Do they cover every aspect?  **4. Conclusion (5 marks)**  **5. References (5 marks)**  **6. Appendices**  **Appendix 1**: User guide: you will need to provide a user manual detailing how to use the marketing application, including instructions for registration, viewing promotions, generating and scanning QR codes, and tracking interactions with promotions.  **Appendix 2**: Code  **Appendix 3**: Test Suites  Your submission must be done via **Moodle**.  **Submit:**   * **A report of 2000 words (or equivalent (+/- 10%)) in word format (.docx).** * **Source codes:** You are required to submit the complete source code for the marketing application. The code must be well-documented and clearly written including (separately)   + **Source code of your application (GUI including QR code scanner and generator)**   + **Source code of your SQLite database** * **5 min Video demonstration of your software**   **IMPORTANT:** The report (item 1 above) should be submitted as **.docx** and **NOT** as a **.zip** file. Any report that will be submitted as part of a .zip file **will not be considered** as it cannot be processed by **Turnitin** system for similarity check.  Submit/add the complete software code for your system, appropriate data and test-cases as a zip file (item 2, 3, and 4 above). |

|  |
| --- |
| **Special Instruction(s)** |
| *Take regular backups of your work. This will enable you to recover quickly should the system fail and also allow you to backtrack if your development goes astray.* |
| *Ensure that the work submitted will execute on University computers.* |

## Assessment: Marking criteria

You need to achieve at least 40% to pass this assessment.

|  |  |  |
| --- | --- | --- |
| **Mark** | **Grade** | **Characteristic s** |
| **0** | Fail | **Plagiarism, collusion, non-pres, name only.** |
| **1-39** | Reassess | **Project not working or extremely limited scope.**  **Report is inadequate. No demonstration. No references.** |
| **40-49** | **3rd, D**  Pass | **Project works but has limited scope and application.**  **Report meet basic requirement correctly but limited, no design or discussion of functionalities.**  **Demonstration not satisfactory. Inadequate references(mainly websites).** |
| **50-59** | **2ii, C**  Satisfactory | **Project is complete and offers considerable application.**  **Report is coherent and organised, some limited evidence of self-criticism concerning deliverables.**  **Sufficient demonstration of functionality, complexity can be improved.**  **Evidence of inclusion and use of references.** |
| **60-69** | **2i, B**  Good | **Project is fully functional, advanced application and complexity.**  **Report has good reflection, is coherent and organised, good integration of academic & practical issues, good evaluation of deliverables.**  **Well explained demonstration showing prototype, working and application.**  **Good quality relevant academic references.** |
| **70-100** | **1st class, A Excellent** | **Project is fully functional, innovative concept, good application scope and advanced complexity for software concepts.**  **Report has good reflection, evidence of elegance, innovation, very good evaluation of deliverables.**  **Well-presented demonstration, all the concepts, ideas, design, implementation, and fully functional software thoroughly explained.**  **Excellent quality relevant academic references (books, journals etc.). Reflective critical analysis built**  **around academic references and experiences from the module.** |

Note that the overall grade will be determined by the application of the School of Computing & Technology Assessment Criteria Grid

#### School of Computing & Technology Assessment Criteria Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mark**  **%** | **Comment** | **Grade & Characteristics** | **Theory &**  **Academic Approach** | **Practice & Deliverables** |
| 0 |  | **Fail** | plagiarism, collusion, non-pres., name only | as theory |
| 1-39 |  | **Reassess:**  ***inadequate but recoverable with effort*** | no understanding, very short, inadequate, factual but little interpretation, lacks coherence, short, errors, misconceptions, coherent but mechanical notes, partial - rudimentary answer, limited interpretation, lack of knowledge of topic, no evidence of background reading,  weak English but some appropriate use of language of topic. | poor effective deliverables, requirements not met, deliverables partially complete, limited response to brief. |
| 40-  49 |  | **3rd, D**  **Pass: Sufficient for award of credit**  ***adequate mainly descriptive approach, fair, limited conceptual or theoretical ability*** | adequate response, demonstration of basic knowledge, relevant content, clear intention communicated, evidence of reading, acceptable minimum level of English for business presentation but may lack precision, some limited analysis / application of  knowledge / theory / weighting of evidence, inconsistent | deliverables meet basic requirement correctly but limited, just adequate but not innovative, interesting or exciting, for higher marks, 45+ just exceeds minimum specification, might be good in some areas but not consistent |
| 50-  59 |  | **2ii, C Satisfactory**  ***Satisfactory with some conceptual ability but lacks good evaluation or synthesis of ideas*** | good response to task, collates info, *satisfactory* analysis & judgement, constructs generalisations based on evidence & opinion, argues clearly, logically & constructs a case, some limited ability to state a personal position, correct English with few imprecise statements | good deliverables, some evidence of good design or execution, coherent and organised product, some limited evidence of self criticism concerning deliverable, some independence, initiative, autonomy, appropriate techniques, integration of knowledge for task |
| 60-  69 |  | **2i, B**  **Good.**  ***Good analysis, evaluation, synthesis, integration & argument.*** | evaluates info. & synthesises generalisations, good ability to state & defend personal position, good analysis & judgement, applies knowledge to new situations, sound on theory, critical, understands limitations of methods, selective coherent & logical approach, well written with clear, correct and precise English | all criteria met to **good** standard, evidence of good design or execution, good integration of academic & practical issues, solid evidence of self critique/evaluation of deliverables, products well organised - documented - coherent. Evidence of independence, initiative, autonomy, creativity, adaptability, resourcefulness.  Integration of knowledge, |
| 70-  79 |  | **First class, A, Excellent.**  ***as above but also stronger evidence of excellent, original, innovative, articulate work*** | very strong ability to state & defend position, uses criteria & weighting in judgements, wide knowledge and theoretical ability, full understanding of possibilities and limitations of methods & theories, *75+* more original, innovative approach, command of critical  positions, lively articulate writing, excellent grasp of material - synthesis of ideas | most criteria met to **high** standard, strong evidence of evaluation of deliverables, *75+:* deliverables excellent - all criteria met in clear and definite manner, evidence of excellent design or execution, elegance, innovation, very good evaluation of deliverables, |
| 80-  89 |  | **Outstanding.**  ***as above but also***  ***authoritative, superlative,***  ***creative*** | **as above but also :-**  seen all possibilities in task, gone beyond accepted conceptual/critical positions, evidence of creative, intelligent, innovative approach consistently & forcefully expressed | **as above but also :-**  all aspects of deliverables superlative  beyond 80% emphasis on theory rather than practice/deliverables |
| 90-  100 |  | **Faultless** | **as for 80-89 but also :-**  all work superlative & without fault | as for 80-89 |