Coursework Assignment Brief

Re-Assessment - Undergraduate

Academic Year 2022-23

Module Title:	Computer Programming	
Module Code:	CMP4266	
Assessment Title:	Computer programme, design, solution and testing.	
Assessment Type:	CWRK	Weighting: 100 %
School:	School of Computing and Digital Technology	
Module Co-ordinator:	OGERTA ELEZAJ	
Hand-in deadline date:	24th July 2023 3:00 pm	
Return of Feedback date and format	Marks and feedback on your work will normally be provided within 20 working days of your submission deadline. See iCity/Moodle for details. http://moodle.bcu.ac.uk	
Support available for students required to submit a re-assessment:	Timetabled support sessions will be arranged for the period immediately preceding the hand-in date	
NOTE:	At the first assessment attempt, the full range of marks is available. At the re-assessment attempt the mark is capped and the maximum mark that can be achieved is 40%.	
Assessment Summary	Computer programme solution and testing. This must include the following parts: **A.** Design documentation submitted online (**Weight 20%**). This must include the two flow chart diagrams. **B.** Programme source code submitted online (**Weight 60%**). **C.** Testing and evaluation inclusive of test cases submitted online. (**Weight 20%**). Further details for each deliverable are below	

IMPORTANT STATEMENTS

Undergraduate Regulations

Your studies will be governed by the BCU Academic Regulations on Assessment, Progression and Awards. Copies of regulations can be found at https://www.bcu.ac.uk/student-info/student-contract

For courses accredited by professional bodies such as the IET (Institution of Engineering and Technology) there are some derogations from the standard regulations and these are detailed in your Programme Handbook

Cheating and Plagiarism

Both cheating and plagiarism are totally unacceptable and the University maintains a strict policy against them. It is YOUR responsibility to be aware of this policy and to act accordingly. Please refer to the Academic Registry Guidance at https://icity.bcu.ac.uk/Academic-Services/Information-for-Students/Assessment/Avoiding-Allegations-of-Cheating

The basic principles are:

- Don't pass off anyone else's work as your own, including work from "essay banks". This is plagiarism and is viewed extremely seriously by the University.
- Don't submit a piece of work in whole or in part that has already been submitted for assessment elsewhere. This is called duplication and, like plagiarism, is viewed extremely seriously by the University.
- Always acknowledge all of the sources that you have used in your coursework assignment or project.
- If you are using the exact words of another person, always put them in quotation marks.
- Check that you know whether the coursework is to be produced individually or whether you can work with others.
- If you are doing group work, be sure about what you are supposed to do on your own.
- Never make up or falsify data to prove your point.
- Never allow others to copy your work.
- Never lend disks, memory sticks or copies of your coursework to any other student in the University; this may lead you being accused of collusion.

By submitting coursework, either physically or electronically, you are confirming that it is your own work (or, in the case of a group submission, that it is the result of joint work undertaken by members of the group that you represent) and that you have read and understand the University's guidance on plagiarism and cheating.

You should be aware that coursework may be submitted to an electronic detection system in order to help ascertain if any plagiarised material is present. You may check your own work prior to submission using Turnitin at the Formative Moodle Site. If you have queries about what constitutes plagiarism, please speak to your module tutor or the Centre for Academic Success.

Electronic Submission of Work

It is your responsibility to ensure that work submitted in electronic format can be opened on a faculty computer and to check that any electronic submissions have been successfully uploaded. If it cannot be opened it will not be marked. Any required file formats will be specified in the assignment brief and failure to comply with these submission requirements will result in work not being marked. You must retain a copy of all electronic work you have submitted and re-submit if requested.

Learning Outcomes to be Assessed:

1. **Describe the fundamental programming principles and the standard programming constructs: repetition, selection, functions, modules and aggregated data.**
2. **Design solutions to programming problems**
3. **Implement those solutions in an imperative programming language by using common programming tools (such as editors and interpreters).**
4. **Use common programming tools and techniques (e.g. IDE, testing APIs and theories) to test and evaluate programs.**

Re-Assessment Details:

Title: Python Coding Practice

Type: Design and Implementation of programming courseworks

Style: *Summative Assessment*

- **Design, development and testing of a computer programme solution**

Part A – Design documentation (weight 20%)

Overview of hospital management systems

In this coursework, students will create a 'prototype' Hospital Management system. This system aims to maintain the medical records of the patient, maintain the contact details of the patient and keep track of the appointment dates. There are three different types of users involved in a hospital management system: admin, doctor and patient. Each of the above plays an important role in the hospital management system performing different tasks. Patients using the system can book online appointments. Their appointments are approved by the admin that also assign a doctor to them.

Use the above activities to draw two flowcharts using the special shapes to represent different types of actions or steps in a process. Link these activities with lines and arrows to show the sequence of the steps, and the relationships among them.

For this part of the assessment, you need to submit two flow chart diagrams in PDF format (total weight 20%). To complete this you might refer to Week 2 learning materials.

- (8%) One diagram should depict the admin log-in process for the hospital management system

- (12%) Second diagram should explain a critical/complicated process of the system e.g., admin approving an appointment requested by a patient and assigning a doctor and informing both the patient and the doctor.

Part B – System development (weight 60%)

Important:
This part of the assessment should not be attempted by students until you are provided with the partial implementation code which you must use as a starting point of your system, and this will be provided to you around Week 8 of this semester. Developing your hospital management system from scratch is not acceptable for this assessment, and may lead into losing substantial amount of marks.

It is critical to use the Python 3.6 or a higher version of Python and "Spyder" IDE to ensure that your submission runs correctly when marked by your tutor.

Description:

Rationale

This programming assignment is to apply the programming principles covered in tutorials and lectures to develop a python software that implements core hospital management solution that is used in by its patients and staff members. The software can be fully implemented using text-based interface, however, to achieve higher marks, a GUI based software can be developed using the Python GUI Programming module namely Tkinter. GUI design and Tkinter programming will be covered during week 10. The aim of the exercises is to enhance a student's experience of programming by applying programming principles to a larger problem of developing a complete application.

There are mainly two reasons behind the selection of the hospital management system as the topic of this coursework. Firstly, the students are familiar with this system, hence, students will spend less time and effort to understand the functions specification of the software they will be developing for this coursework. Hence, most of their time will be devoted to the design, development and testing of the system by applying the programming knowledge and skills they learnt throughout this module. Secondly, GUI based programs make it easier to interact with the developed system and demonstrate a direct relationship between the user interaction and the system functions and data. Also, students will learn about how a product works entirely from the user's (or customer's) perspective and not from just a developer prospective. Hence, they will need to develop a user-friendly GUI.

Objective - Sample prototype application

The objective of this assessment is to develop a python software that implements a core hospital management solution that is used in hospitals by its patients and hospital admins. A partial implementation of a prototype system will be developed during the practical sessions in the PC labs. This prototype will be developed to include basic system functionalities such book an appointment and display patients data by doctors. The main class will allow to add a number of patients accounts at the start of the software. At the initial stage, all patient's data will be hard coded.

However, for students to achieve higher marks, they should enhance the system functionalities by using text files to store the patient's related information.

In the system we have one active role that is the **Admin**. Doctors and Patients are passive objects. In this system only Admin dashboard will be developed and Patients and Doctors will act as data storage instances.

Assessment specifications with a detailed marking scheme:

To achieve a mark to maximum of 40% of the total marks for the System Development

The application **must** implement **all** the following:

Create the necessary classes and functions which allow different types of users to perform the following tasks

Admin

- Admin Login
- Register/view/update/delete doctor

- View patient
- Can view and approve appointment requested by the patients
- Can assign doctor to a patient

Doctor

- View their patient details assigned to them by admin
- View their appointments

Patient

- Apply to be admitted in the hospital
- Book an appointment
- View the assigned doctor
- Check the status of their appointment (approved, pending)

To achieve a mark of 41% to maximum of 50%

The application **must** implement **all** the above and the following:

Create the necessary classes and functions which allow admins to perform the following tasks:

- Discharge a patient account i.e. remove patient from the system when the treatment is done
- Admins can view the discharged patient list
- Update admin own information i.e. name and address
- Print all doctor details
- Print all patient details

To achieve a mark of 51% to maximum of 70%

The application **must** implement **all** the above and the following:

- Patients have names, symptoms, age, mobile, address etc.
- Patients of the same family are grouped together by Admin
- The hospital system should be able to store and load all patients' data from and into a file

To achieve a mark of 71% to maximum 80%

The application **must** implement **all** the above and the following:

- Relocating patients from one doctor to another.
- Admins can request a management report. This should show the following information:

  i) Total number of doctors in the system
  ii) Total number of patients per doctor
  iii) Total number of appointments per month per doctor
  iv) Total number of patients based on the illness type.

To achieve a mark of 80% and over

- View diagrams of the reports from i, ii, iii, and iv.
- Development of a suitable Graphical User Interface (GUI) to perform all the above functions.

Note: students need to submit their system alongside the necessary objects to test their software by the tutors when marking.

Part C – Testing and evaluation (weight 20%)

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Students are expected to perform White Box Testing (also known as Clear Box Testing). White box testing involves looking at the structure of the code. Since you know the internal structure of a system, tests can be conducted to ensure that the internal methods and operations perform according to the specification. Students must choose inputs to exercise paths through the code and determines the appropriate outputs. Good testing means your system is free of logical and run time errors.

For this part, you need to develop a test plan that includes test cases based upon the system design and implementation of your Hospital Management System.

- We expect around 10 unique and valid test cases for your system.
- Testing and evaluation mechanism and methods will be explained during Week 10 when we will cover “Debugging and Test Plan”.
- Testing and evaluation will only be done when the 1st version of your software has been implemented.

All test tables should be submitted in PDF format

Additional information: The following links provides further information about the programming language used for teaching this module:

Essential Book:

Gaddis, T. (2015). Starting Out with Python. 3rd ed. Pearson Education

Recommended Tutorials:

Portable Python (http://www.portablepython.com)

Python Website (http://www.python.org)

Python Official Documentation (https://docs.python.org/3/)

Python Official Tutorial (https://www.python.org/about/gettingstarted/)

Recommended Readings:

- Downey, A. (2015). Think Python How to Think Like a Computer Scientist. 2nd ed. O'Reilly Media
- Shaw, Z. (2013). Learn Python the Hard Way. 3rd ed. Addison Wesley

- Spraul, V. A., (2012) Think Like a Programmer: An introduction to Creative Problem Solving, William Pollock

For advice on writing style, referencing and academic skills, please make use of the Centre for Academic Success: https://icity.bcu.ac.uk/celt/centre-for-academic-success

Workload: It is expected that the students should take around 60/75 hours (in addition to the regular weekly teaching hours) of programming effort to design and develop an application as specified in this assessment brief.

Transferable skills

The assessment help students to develop various transferable skills which are necessary to obtain a decent job in the future.

Problem Solving

Being a programmer involves the ability to identify real-world issue and coming up with a technological solution to address it. This requires having strong analytical skills to understand the issue and evaluate different solutions in order to find the one that best fits the problem.

Ability to express your thoughts effectively

Using flowcharts provide an effective way of communicating the logic of a system to all people involved in the project regardless of their background and expertise. They can serve as an excellent bridge between software developers and end users.

Project evaluation skills

It is the process that critically examines a program. It involves collecting and analyzing information about program activities, characteristics, and outcomes in relation to customer specifications. Its purpose is to make judgements about a programme that you have developed, to improve its effectiveness and inform programming decisions.

Time management and planning

Design documentation serve as the basis for the estimating, scheduling, and validating efforts.

Additional information

It is recommended that you start this assignment as early as possible in Week 8 and that you build your report week on week. Ensure you get valuable feedback on draft work as you build your report. Do not ask for feedback on the day before submission and instead plan your workload pertaining to this assessment throughout the module.

Marking Criteria:
Table of Assessment Criteria and Associated Grading Criteria

	0 – 39% Fail	40 – 49% Third	50 – 59% 2:2	60 – 69% 2:1	70 – 79% First	80 – 100% First
Criterion A	**LO1 - Design solutions to programming problems**					
Mark: 20%	A basic but relevant attempt at understanding and expressing a possible internal design solution for the problem domain. Presented in text only, no attempt for diagrammatic presentation.	A reasonable and relevant attempt at understanding and expressing a possible internal design solution for the problem domain. Diagrammatic presentation attempted, possibly incomplete and inaccurate.	A good and relevant attempt at understanding and expressing a possible internal design solution for the problem domain. Diagrammatic presentation attempted. Possibly incomplete but accurate, or complete design attempted but inaccurate.	A very good attempt at understanding and expressing a possible internal design solution for the problem domain. Complete and accurate diagrammatic presentation.	An excellent expression of a possible internal design solution for the problem domain. Complete and accurate diagrammatic presentation accompanied by clear documentation of the design.	An excellent expression of a possible internal design solution for the problem domain. Complete and accurate diagrammatic presentation accompanied by clear documentation of the design. Discusses the wider issues in the problem domain that may impact the proposed design.
Criterion B	**LO2 - Implement solutions in an imperative programming language by using common programming tools (such as editors and interpreters).**					
Mark: 60%	Significantly incomplete program. Does not compile and execute.	Program compiles and demonstrate only few basic features of the design.	Program demonstrates most of the basic features of the design	Program demonstrates all of the basic features of the design and some other advanced features of the design.	Program demonstrates all the basic and advanced features of the design.	Program demonstrate all the basic and advanced features of the design. In addition demonstrates some additional features relevant to the problem.
Criterion C	**LO3 Use common programming tools and techniques (e.g. editors, testing APIs and theories) to test and evaluate programs**					
Mark: 20%	Little or no testing of the program. No rigour in	Weak testing that does not adequately	Basic testing of the progra m show ing implemented	Adequate testing of all program	Excellent testing of all program	Excellent testing of all program features

Submission Details:

Format:

All work should be submitted on Moodle as a single zip file. The zip file should contain the following:

- Completed software implementation checklist (Compulsory).
- Design documentation which includes two flow chart diagrams (to be submitted as PDF files)
- All of your source code and associated project files.
- Testing and evaluation inclusive of test cases submitted as PDF files.

The zip file should be structured so that the location of its contents is clear and unambiguous. Please submit only one version of your work. Also, the zip file should be named using the following format StudetName_studentID.zip

For example: **ShadiBasurra_123456789.zip**

PLEASE MAKE SURE THAT YOUR APPLICATION WILL WORK ON THE UNIVERSITY'S MACHINES. DO NOT RELY ON THE FACT THAT IT MAY WORK ON YOUR OWN LAPTOP/PC AND THEREFORE IT SHOULD WORK ELSEWHERE.

Regulations:

- Re-sit marks are capped at 40%

Full academic regulations are available for download using the link provided above in the IMPORTANT STATEMENTS section

Late Penalties

- **If you submit a re-assessment late then it will be deemed as a fail and returned to you unmarked.**

Feedback:

Feedback on your submissions should be given electronically via Moodle.
Marks and Feedback on your work will normally be provided within 20 working days of its submission deadline via Moodle.

Where to get help:

Help and support to complete the assessment will be provided during the followings:

- Assessment support sessions during teaching weeks

Students can get additional support from the library for searching for information and finding academic sources. See their iCity page for more information: http://libanswers.bcu.ac.uk/

The Centre for Academic Success offers 1:1 advice and feedback on academic writing, referencing, study skills and maths/statistics/computing. See their iCity page for more information: https://icity.bcu.ac.uk/celt/centre-for-academic-success

See also the My Assignment Planner tool: http://library.bcu.ac.uk/MAP2/freecalc-mail/

Fit to Submit:

Are you ready to submit your assignment – review this assignment brief and consider whether you have met the criteria. Use any checklists provided to ensure that you have done everything needed.

Use checklist provided on Moodle