CPT109 C Programming and Software Engineering 1 – ASSESSMENT 3

Assessment Number	3
Contribution to Overall Marks	35%
Issue Date	08/11/2021
Submission Deadline	17/12/2021 at 17:00 (5pm)

Assessment Objective

This assessment aims to evaluate students' ability to develop a significant software solution to a real-world problem by working as a member of a team. Your team will be given a vague specification and is expected to deliver a software product in the C programming language, which meets the specifications before the due date. This size and type of the project is suitable for development in modular format and so teams are encouraged to devise program structures that allow various parts of the code to be developed independently by each team member. Being a team player means you are expected not only to apply the knowledge gained during the lectures, laboratory classes and assignments to specify, design, implement, test and document your own code, but also to cooperate with your teammates so that the whole project will be delivered on time with good quality.

Grouping

There are 360 students enrolled in this module, and you will be divided into groups consisting of 5 students (i.e. 72 groups). Groups will be formed in two stages as follows: Firstly, students will be given the option to choose their own group members. Students failing to form a group will then be randomly assigned to a group. Randomly formed groups will contain students with a range of ability based on their performance in previous assignments. Each group will then be randomly assigned one of 5 projects. Students wishing to form their own group should submit a **hardcopy** of the form provided (The **group formation form** is part of the Assessment 3 download package) having all details filled in and having been **signed by all** group members.

Final Deliverables

Each group should submit the following (only 1 submission per group):

1. A report (PDF file), based on the Software Development Process applied throughout this semester:
   a) Problem Statement (Specification: formulate the problem generally) (10%).
   b) Analysis: interpret the vague software requirements provided by your design brief and determine a very clear specification for your software design. (10%)
   c) Design: explain how your program is structured in terms of functional blocks and describe what each block does. A flow diagram may be useful here showing how functions interconnect and what data is passed between them. (10%)
   d) Testing: explain how the program has been tested and verified. Ensure any usernames or passwords required to test the program are listed in a table. (10%)

2. All C source code (.c files) and the final executable demonstration file (.exe). The source code must be appropriately commented. Include any data files needed for execution. (50%)

3. **Contribution form** (The contribution form is part of the Assessment 3 download package). The group should agree on the percentage of contribution of each member to each section and submit one copy of the form **signed by all** members.

NOTE: This may lead to different marks for different members of the same group. If necessary, the module leader may call group members for a short oral test.

Submission Procedure

All the above-mentioned files/deliverables (report, source code, executable, manual, contribution form, must be compressed into a single file (.rar, .zip or .7z). The group leader (identified on the group formation form) must submit this single compressed file on Learning Mall using his/her account.

NOTE: Each group should only submit **ONE** copy on Learning Mall. Make sure your report has a title page and ensure **ALL** group members' names and ID numbers are on it.

Marking Scheme

The general marking scheme is shown as follows:

Documentation	**(50%)**
Overall Report Quality	10%
Problem Specification	10%
Analysis	10%
Algorithm Design	10%
Testing	10%
Coding	**(50%)**
Implementation/coding style	40%
Robustness	10%

Please refer to the file "CPT109 Marking Guidelines for Assignment 3" in the Assessment 3 download package on Learning Mall for details on marking descriptions.

Important notes:

University late submission policy will be applied.

University Academic Integrity policy will be applied rigorously.

You can find information relating to both of these in ebridge.

General Guidelines

The project descriptions are deliberately given in the form of simple customer specifications, which (as in the real world) are incomplete and often ambiguous, rather than a set of exact functional specifications. The group members should work methodically together (as the developers in a real world software project would) to:

1. Analyse and formalise the customer specifications (at this stage, the various design choices and the software features can be subject to the group's creativity).
2. Design and decompose the functional and programmatic aspects of the problem and allocate constituent tasks to each group member. You are expected to use a top-down design which can then be modularised so that the tasks for each member can be clearly determined.
3. Implement the product with frequent meetings to report progress and decisions to each other and re-evaluate the agreed courses of action.
4. Implement test procedures, debug and correct the program. Each program module should be independently testable. Testing of each module and the program as a whole should be performed.
5. Finalise the deliverables.

The specifications are only basic and most of the design choices should be made in your group meetings. The systems described within the different projects have a variety of different features and the disambiguation of the customer specifications can be based on the student's logic and real life experience.

Assessment will be based on whether the product/program offers reasonable functionality and features (for the group size, allocated time and project difficulty), its design quality, flexibility, robustness, software bugs and other stated deliverables.

If the group cannot implement all of the system features mentioned, it is better to have a few features fully working without run-time crashes than none of the required features working properly due to bugs or disrupting ripple effects between modules in the project. However, the corresponding marks deduction will be applied depending on the missing features.

If any group issues arise during the project, it is important to raise these with the module leader as soon as they do so that resolutions can be found quickly. It is important to work with your group, and to share the work accordingly. All group members should be responsible for some of the coding. Individuals may not produce your own work for submission independent of the group.

Project A: Bank Information System

Overall description:

Your team is employed by a bank to implement a system to manage the banking affairs.

Customer specifications:

The implementation of the banking system should be able to provide facilities to:

- Register a new customer and store details such as name, address, telephone number, 6-digit personal identification number (PIN) (security code for using the card) as well as some extra information e.g., type of identification presented for joining the bank, etc.
- Store and manage customer account information. Customers can have one or more accounts and each account is uniquely identifiable by an account number generated by the system.
- Collect and display bank statistical information for the bank manager e.g. number of customers, number of accounts.
- Allow the following banking activities:
  - Display current account balance
  - Allow withdrawals from an account.
  - Register a deposit

System Users

The system should be able to provide functionality for different users listed below:

- **Manager** who will be able to:
  - View banking statistics (as described above).
- **Bank clerk** who will be able to:
  - Add/delete/edit accounts for an existing customer.
  - Make deposits to a customer's account.
- **Customer** who will be able to:
  - Access their account information and perform banking activities **except** the deposit of money.
  - All banking activities require the customer to enter their PIN number.

Project B: Parking Management System

Overall description:

Your team is employed for the implementation of a parking payment system in a university car-park. The parking space consists of:

- A 12x12x5 multi-story carpark, where each of the 5 floors is a 12x12 rectangular grid of car parking spaces.
- A 30x2 grid of e-bike parking spaces each with an electrical recharge plug is provided.
- A 30x4 grid of e-bike parking spaces with no recharge facility (cost should be lower than that for a recharging space).

Customer specifications:

The implemented parking system should be able to provide facilities to:

- Maintain a database of registered customers including their personal details: name, university ID number, phone number, account balance and staff or student status.
- Registered customers should be able to pre-pay for their parking or to accumulate charges (up to a pre-set amount).

System Users

The system should be able to provide functionality for different users listed below:

- **Administrator** who will be able to:
  - Add/delete/edit customer's personal details
  - Add money to a customer's account
  - Set parking charges
- **Entry/Exit system worker** who will be able to:
  - Record the entry of a registered customer.
  - Allocate a randomly chosen parking space to each car as it enters. For an e-bike, the customer should be asked if they would like a space with/without recharging and then a randomly chosen space is allocated.
  - View current car-park occupation and number of free spaces.
  - Free a previously allocated slot
  - Charge the leaving customer accordingly.

Project C: Library Information System

Overall description:

The university library needs a new electronic rental system and your team is employed to build it.

Customer specifications:

The implemented system should be able to handle the basic operations of a library including the following features:

- Catalogue the library books; they should be stored in an indexed order e.g. by author name or shelf-mark.
- The information about each book title should include: author(s), title, ISBN, subject, loan type (normal, short loan, no-take-out), shelf-mark, loan status, number of copies, etc.
- Provide search functionality so that any user can find a book

System Users

The system should be able to provide functionality for different users listed below:

- **Administrator** who should be able to:
  - Add and edit book information
  - Register new library users including name, university ID number, phone number, staff or student status, book loans, charges.
  - Record the return of a book from a borrower and calculate any charges
- **Borrower (Staff or Student)** who should be able to:
  - Ability to borrow books
  - Ability to edit their personal details
  - Ability to renew their current loans for a pre-set number of times.

Project D: Coffee Shop Ordering System

Overall description:

Your team is employed by a new coffee shop to develop the software for an in-store terminal where customers can order their drinks.

Customer specifications:

The implemented ordering system should be able to provide facilities to:

- Allow the various coffee based drinks on the menu (e.g. latte, mochaccino etc.) to be ordered where each coffee can be customised by: coffee bean type, milk type, hot/ice and has additional options including extra shot of coffee and number of sugars.
- Determine the cost of the drink ordered.
- Register and edit customer accounts, identifiable by a unique account number, which can be pre-charged with money. Customer information should include name, telephone, order history, available balance.

System Users

The system should be able to provide the functionality for different users listed below:

- **Shop manager** who will be able to:
  - Edit the drinks and options menus.
  - Set/alter charges for each drink/option.
  - Add/remove/edit customer information.
  - Provide statistics related to coffee sales.
- **Customer Self Service** will be able to:
  - Login to account and place an order.
  - Review order history.
  - Deposit money on the account.

Project E: Hotel Management Information System

Overall description:

Your team is employed by the conference centre hotel to implement a software system responsible for the overall management of room booking and customer records.

Customer specifications:

The implemented hotel systems should be able to provide facilities to:

- Manage bookings for 60 rooms (10 per floor) and three classes of room (**, ***, ****). Each room is assigned a single price class.
- Manage customer accounts. Customer accounts are identifiable by a unique account number.
- Offer hotel business statistics e.g. numbers of hotel guests.

System Users

The system should be able to provide functionality for different users listed below:

- **Manager** who will be able to:
  - Set/amend classes for each room and the price per class.
  - Manage a customer database (add/edit/remove customers)
  - View hotel business statistics e.g. number of rooms booked, total income.
- **Receptionist** who will be able to:
  - Register a booking.
  - Edit booking details e.g. period of stay or room class.
  - Check-out customers by calculating charges.

CPT109 Assessment 3 - Group Formation

Please fill the table below, put a cross '**x**' in the final column to indicate the team leader. Ensure all members have signed the form.

Submit to the Red Box outside EE514 before 5pm on Friday 5th Nov. 2021

ID Number	Name (English only)	Email Address	Indicate team leader

Signed by all members: _____

_____

_____

_____

_____

CPT109 Assessment 3 - Contribution Form

Group Number:

Name	ID Number	Specification & Analysis (%)	Design (%)	Coding (%)	Testing (%)	Report (%)
1.						
2.						
3.						
4.						
5.						
	Total	**100**	**100**	**100**	**100**	**100**

Signed by all members: _____

_____

_____

_____

_____