cct | College Dublin
Computing • IT • Business

CCT College Dublin Continuous Assessment

Programme Title:	*H.Dip. in Computing*		
Delivery Mode:	FT		
Cohort Details:	Feb 2025 cohort		
Module Title(s):	Algorithms & Constructs Software Development Fundamentals		
Assignment Type:	*Integrated / Individual*	**Weighting(s):**	Algorithms & Constructs – 50% Software Development Fundamentals – 55%
Assignment Title:	*System Modelling & Build*		
Lecturer(s):	*Ken Healy / Taufique Ahmed*		
Issue Date:	*28th Oct 2025*		
Submission Deadline Date:	*Saturday 29th Nov at 11.59pm*		
Late Submission Penalty:	Late submissions will be accepted up to **5** calendar days after the deadline. All late submissions are subject to a penalty of **10%** of the mark awarded. Submissions received more than 5 calendar days after the deadline above **will not** be accepted and a mark of 0% will be awarded.		
Method of Submission:	**This assignment is submitted via Moodle.**		
Instructions for Submission:	**Algorithms & Constructs** A GitHub repo must be created. The Netbeans project and report Word document must be put into a GitHub repo for version control. The lecturer must be added to the repository as a collaborator. The GitHub repo link will be added at the end of the report. There should be at least 10 to 15 commits throughout the time worked on the project. Submit your Netbeans .java files individually, and make sure all files contain the same package name, CA_2. Compressed files, such as ZIP or RAR will not be marked! If your submission does not load or run, then you may receive a zero mark! Include a section (max 500 words) in your Report submission that clearly explains your choices for the sorting & searching algorithms that you have used. **Software Development Fundamentals**		

	Your design diagrams, user stories, etc. must be submitted in one document in **PDF format** to the SDF Moodle page. You do not need to include this in your GitHub Repo. There will be an IN-CLASS Q&A held on Tues 2nd Dec. You may be asked to explain any part of your submission. Late submissions will be required to attend a separate Q&A in week commencing 8th Dec.
Feedback Method:	**Results posted in Moodle gradebook**
Feedback Date:	*Following release of approved results by the college*

Assessment Outline

Description of Assessment Task

You have been tasked to **model and create** a working program to demonstrate an “Organisation” as a prototype.

Core requirements:

- Must contain 3 parent Classes, i.e. Manager, Department and Employee
    - More marks will be awarded for having at least three different types of each

- Console menu (See Algorithms & Constructs section for minimum requirements)
- User input and validation
- Generate random (but appropriate) data
- Implement custom searching and sorting algorithms

Specific Restrictions

This is a command line program, and the use of graphic user interfaces is **not** allowed (i.e. NO JavaSwing, JavaFX, GUI builders, etc.).

The use of dependencies – such as Maven or Gradle – is **not** allowed.

Code quality standards apply - marks may be lost for poor programming practices, such as non-descriptive variable names, poor/no comments, etc.

Note that these are the **minimum** requirements for passing and that more marks may be awarded for better implementations.

Software Development Fundamentals: Modelling and Testing

Create **FIVE** Use Cases for the system. These should be clear and concise. Each Use Case narrative should be accompanied by an appropriate Use Case Diagram. Document where in your programme code [Algorithms & Constructs] the use cases have been implemented.

[20 Marks]

You are required to use at least **TWO DIFFERENT** kinds of UML modelling techniques to create models of the software system. [**Note:** These cannot be use-case diagrams!]
Note: Higher marks will be awarded for modelling features that could be included in a future version of the system. These should be included in your diagrams.

[30 marks]

Provide a set of SENSIBLE requirements for your system. These should be divided into functional **and** non-functional requirements. Make sure that it is clear why each requirement has been included. You should aim to have at least 5 functional and 5 non-functional requirements.

[20 marks]

Create a set of FIVE user stories, including acceptance criteria, using the use cases and UML models that you have provided.

[15 Marks]

In addition, include THREE **unit** tests that you would employ, demonstrating your knowledge of good testing practice.

[15 marks]

NOTE: A Q&A Session will be held in class after the submission deadline. You can expect to be asked to explain at least 1 of your diagrams, user stories or unit tests. If you are unable to answer questions satisfactorily then your mark will be reduced and you may be asked to attend a longer meeting to determine if the work is your own.

Algorithms and Constructs

You will be required to choose one of the following three organizations: **School, Bank** or **Department Store** Your role will be to analyze and assess the unique challenges, operations, and organizational structure of your chosen entity. You will need to explore its core functions, the environment in which it operates, and the roles of key staff.

Terminal Menu Examples: The user will be required to select from a number of options from the menu, use **Enums for the Menu** to iterate through the options available from the list below, this is required for all sections: **Sort a Dummy List of People, Search in the List and Return Relevant Information, Allow for New User Input (Name, ManagerChoice, and Department), Generate Random People with Manager Types and Departments.**.

Command line Example 1:

```
Please enter the filename to read:

Applicants_Form.txt

File read successfully

Do You wish to SORT or SEARCH:
1. SORT
2. SEARCH
3. ADD RECORDS
4. EXIT

1

SORT selected
```

Command line Example 2:

```
Please select an option from the following:

1. Add Employee
2. Generate Random Employee
```

```
1

Please input the Player Name:

John Joe

Please select from the following Management Staff:

1. Head Manager
2. Assistant Manager
3. Team Lead

2

Please select the Department:

1. Customer Service
2. Foreign Exchange
3. HR

2

“John Joe” has been added as “Assistant Manager” to “Foreign Exchange” successfully!
.
.
```

Sort a Dummy List of People: You have been given a file named **Applicants_Form.txt**, which contains a list of randomly ordered names collected from form submissions. Your task is to sort this list in alphabetical order and display only the first 20 names on the screen. Sorting the names properly ensures that the data is well-organized, easy to search, and efficiently updated when needed.
Choose an appropriate **RECURSIVE sorting algorithm** to sort the list. Justify your choice by explaining why this **SORTING** algorithm is the best fit compared to other alternatives. **Do NOT** provide a definition of how the algorithm works and focus on why you selected it.
[20 Marks]

Search in the List and Return Relevant Information: Your program should allow users to search for a person by name from the sorted list and return the following details, such as Manager Type (e.g., Senior Manager, Team Lead, etc.) and Department Name (e.g., IT, HR, Finance, etc.)
Choose an appropriate searching algorithm that efficiently finds the requested name in the list. Justify your choice by explaining why this algorithm is the best fit compared to other alternatives. Do NOT provide a definition of how the algorithm works and focus on why you selected it.
[15 Marks]

Allow for New User Input (Name, Manager Choice, and Department): The program should be interactive, enabling users to input new data. Users should be able to enter a name, choose a valid Manager Type, and select an existing Department for that person. This data should then be added to the list and appropriately stored in the program's memory. To maintain data integrity, user inputs must be validated to ensure that the selected Manager Type and Department must exist. Once a new entry is added, the program should update the list accordingly and display all newly added records in the terminal.
[20 Marks]

Generate Random People with Coach Types and Teams: In this part, you will implement a mechanism to generate random individuals and assign them a valid Manager Type and Department. This feature will be useful for testing and populating the system with initial data. To maintain consistency, the randomly generated data must align with the predefined Manager Types and Departments, avoiding any conflicts. Once generated, all employees, including both pre-existing and newly created ones, should be listed in the terminal to ensure transparency and verify correctness.
[30 Marks]

Enums for the Menu: Use **ENUMS** to define menu options, making the user interface more structured, organized, and easier to manage.
[15 Marks]

Assessment Requirements

All assessment submissions must meet the following minimum requirements:

- Be submitted in the format outlined in the assignment summary table.
- Comply with the word count limits specified (where this is relevant).
- Be submitted by the deadline date specified or be subject to late submission penalties.
- Be submitted via Moodle upload. Any other submissions will not be graded.
- Use Harvard Referencing when citing third party material.
- Be the student's own work.
- Include the CCT assessment cover page.

Learning Outcomes:

This assessment addresses the following module learning outcomes for each module, as below:

Algorithms & Constructs

- Evaluate and select appropriate Algorithms and types of Data Structures for a given problem, considering complexity and system resource constraints (Linked to MIPLO 1 and 3).

- Use recursive algorithms as a means of Problem solving (Linked to MIPLO 2, 3 and 7).

Software Development Fundamentals

- Determine system requirements using process modelling techniques.
- Compare and contrast process models in the development of software systems.
- Test the practical applications of the design process incorporating Use Cases.

Statement of Acceptable Use of Artificial Intelligence

Acceptable and Unacceptable Use of AI

- The use of generative AI tools (e.g. ChatGPT, Dall-e, etc.) is permitted in this assignment for the following activities:
  - Brainstorming and refining your ideas;
  - Fine tuning your research questions;
  - Finding information on your topic;

- The use of generative AI tools **is not permitted** in this course for the following activities:
  - Impersonating you in any context
  - Generating program code of any kind
  - Writing a draft of your assignment report
  - Writing or rewriting entire sentences, paragraphs or any element of your assignment.

- You are responsible for the information you submit based on an AI query. Your use of AI tools must be properly documented and cited.

- Any assignment that is found to have used generative AI tools in an unauthorised way will be subject to college disciplinary procedures as outlined in the QA Manual.

- When in doubt about permitted usage, please ask for clarification.

Grading Criteria

Software Development Fundamentals

Description	Weighting
Use Cases / Diagrams [FIVE required – Diagram + Use case Specification for each]	20
UML Model 1 [NOTE: Must *not* be a Use-case diagram!]	15
UML Model 2 [Must be DIFFERENT from Model 1. Use-case diagram not allowed!]	15
Requirements List [Must include BOTH functional and Non-Functional requirements.]	20
Unit Tests [THREE required. Should follow a Unit Test template. NO CODE required!]	15
User Stories & Acceptance Criteria [FIVE required. Stories should follow standard structure. Sensible criteria are needed for each! Remember these are different from tests!]	15
TOTAL (Software Development Fundamentals)	**100**

Algorithms and Constructs

Description	Weighting	Mark
Recursive Sorting Algorithm - Implementation & Rationale (i.e. a clear reasoning is provided for why this algorithm is chosen)	20	
Searching Algorithm - Implementation & Rationale (i.e. a clear reasoning is provided for why this algorithm is chosen)	15	
New User Input (Name, Coach Choice, and Team) & Validation	10	
Generating Random Data [Code must be commented properly!] Use appropriate Data Structure.	30	
Menu utilises enums in a clear & sensible way. Code is well-commented to provide clear explanations of the implemented logic and reasoning behind key decisions.	15	
Total	**100**	

The Irish Grading System

The grading system in CCT is the QQI percentage grading system and is in common use in higher education institutions in Ireland. The pass mark and thresholds for different grade bands may be different from what you have experienced in the higher education system in other countries. CCT grades must be considered in the context of the grading system in Irish higher education and not assumed to represent the same standard the percentage grade reflects when awarded in an international context.

Please review the CCT Grade Descriptor available on the module Moodle page for a detailed description of the standard of work required for each grade band, and review the marking criteria outlined in this assignment brief for a breakdown of the marking criteria for this specific assignment.

Additional Information

- Lecturers are not required to review draft assessment submissions. This may be offered at the lecturer's discretion.
- In accordance with CCT policy, feedback to learners may be provided in written, audio or video format and can be provided as individual learner feedback, small group feedback or whole class feedback.
- Results and feedback will only be issued when assessments have been marked and moderated / reviewed by a second examiner.
- Additional feedback may be provided as individual, small group or whole class feedback. Lecturers are not obliged to respond to email requests for additional feedback where this is not the specified process or to respond to further requests for feedback following the additional feedback.
- Following receipt of feedback, where a student believes there has been an error in the marks or feedback received, they should avail of the recheck and review process and should not attempt to get a revised mark / feedback by directly approaching the lecturer. Lecturers are not authorised to amend published marks outside of the recheck and review process or the Board of Examiners process.
- Students are advised that disagreement with an academic judgement is not grounds for review.
- For additional support with academic writing and referencing students are advised to contact the CCT Library Service.
- For additional support with subject matter content students are advised to contact the CCT Student Mentoring Academy
- For additional support with IT subject content, students are advised to access the CCT Support Hub.