cct | College Dublin
Computing • IT • Business

CCT College Dublin Continuous Assessment

Programme Title:	HDip in Computing		
Cohort:	Sep 2023 FT		
Module Title(s):	Software Development Fundamentals Project Skills and Professionalism Algorithms & Constructs		
Assignment Type:	Individual	**Weighting(s):**	Software Development Fundamentals - 55% Project Skills and Professionalism - 60% Algorithms & Constructs - 60%
Assignment Title:	Rugby Club System		
Lecturer(s):	Kayoum Khbuli James Garza		
Issue Date:	15th April 2024		
Submission Deadline Date:	12th May 2024		
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:	**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:** Report in Word format (.doc/.docx) only. All diagrams should be included.		

	Project Skills: Report in Word format ONLY. (the word count is ~1,200) Diagram in Word format ONLY. Presentation: Create a recorded presentation and save this on a cloud resource (e.g. YouTube). Do NOT submit video files to Moodle! Submit a .pptx or PDF format (ONLY!) of the slides that accompany your presentation. Include ONE extra slide that contains the URL of the recording you have made. This slide should not form part of your recording. The presentation submitted must be the SAME as the one given that you have recorded!
Feedback Method:	**Results posted in Moodle gradebook**
Feedback Date:	Following the release of official results by the Board of Examiners.

Learning Outcomes:

Please note this is not the assessment task. The task to be completed is detailed on the next page. This CA will assess student attainment of the following minimum intended learning outcomes:

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.

Algorithms and 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).

Project Skills and Professionalism

1. Analyse problems and undertake research, identifying and drawing upon valid sources, to formulate proposals or solutions (linked to MIPLO 6, 7 and 8)

2. Evidence of critical thinking and creativity in solving problems, designing, implementing and evaluating solutions (linked to MIPLO 1, 2, 3, 4 and 8)

3. Write, present and defend proposed solutions to industry-related challenges, drawing on the theoretical knowledge gained and the tools and strategies learned in the other modules of the programme, as well as insights and analysis secured from research (linked to MIPLO 6 and 8)

5. Critically analyse problem-solving approaches (employed by self or others) and propose alternatives (linked to MIPLO 4, 5 and 6)

Attainment of the learning outcomes is the minimum requirement to achieve a Pass mark (40%). Higher marks are awarded where there is evidence of achievement beyond this, in accordance with QQI *Assessment and Standards, Revised 2013*, and summarised in the following table:

Percentage Range	**CCT Performance Description**	**QQI Description of Attainment**	
		Level 6, 7 & 8 awards	**Level 9 awards**
90% +	Exceptional	Achievement includes that required for a Pass and in **most** respects is significantly and consistently beyond this	Achievement includes that required for a Pass and in **most** respects is significantly and consistently beyond this
80 – 89%	Outstanding		
70 – 79%	Excellent		
60 – 69%	Very Good	Achievement includes that required for a Pass and in **many** respects is significantly beyond this	Achievement includes that required for a Pass and in **many** respects is significantly beyond this
50 – 59%	Good	Achievement includes that required for a Pass and in **some** respects is significantly beyond this	Attains all the minimum intended programme learning outcomes
40 – 49%	Acceptable	Attains all the minimum intended programme learning outcomes	
35 – 39%	Fail	Nearly (but not quite) attains the relevant minimum intended learning outcomes	Nearly (but not quite) attains the relevant minimum intended learning outcomes
0 – 34%	Fail	Does not attain some or all of the minimum intended learning outcomes	Does not attain some or all of the minimum intended learning outcomes

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.

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 experience of 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.

Assessment Task

Students are advised to review and adhere to the submission requirements documented after the assessment task.

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

Core requirements:

- Coach, Team, and Player classes
  - 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

Write **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.

[15 Marks]

You are required to use at least **TWO DIFFERENT** kinds of UML modelling techniques to create models of the software system.
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 report.

[30 marks]

Justify the specific models that you chose by comparing these to alternative modelling techniques and explaining why your choices were most appropriate in the context of this system. [Note: marks will not be awarded for a generic justification – you must relate this to the system you are designing!]

[25 marks]

Create a set of FIVE user stories with acceptance criteria and acceptance tests using the use cases and modelling techniques from above. In addition, include FIVE unit tests that you would employ, demonstrating your knowledge of good testing practice.

[30 marks]

Description	Weighting	Mark
Use Cases / Diagrams	15	
UML Model 1	15	
Justification of Model 1	10	
UML Model 2	15	
Justification of Model 2	10	
Comparison with Alternative Model(s)	5	
Unit Tests	15	
User Stories, Acceptance Criteria & Tests	15	

Algorithms and Constructs

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, Coach Choice, and Team), Generate Random People with Coach Types and Teams**.

Command line Example 1:

```
Please enter the filename to read:

Club_Form.txt

File read successfully

Do You wish to SORT or SEARCH:
SORT(1)
SEARCH(2)

1

SORT selected
```

Command line Example 2:

```
Please select an option from the following:

Add Player(1)
Generate Random Player(2)

1

Please input the Player Name:

John Joe

Please select from the following Coach Staff:

Head Coach(1)
Assistant Coach(2)
Scrum Coach(3)

2

Please select the Teams:

A Squad(1)
B Squad(2)
Under-13 squad(2)

2
```

“John Joe” has been added as “Scrum Coach” to “B Squad” successfully!
.
.

Sort a Dummy List of People: You will have a list of random names in a file called **Club_Form.txt** gathered from the form submissions of interested applicants to the Rugby Club; this list should be sorted in alphabetical order. The output of the sort should be displayed on the screen [show the first 20 records only]. The sorting of the list will ensure that the data is organised and can be efficiently searched and updated.
You will need to select an appropriate **RECURSIVE sorting algorithm** AND **provide a clear explanation** for why you chose this algorithm and not an alternative. Do not provide definitions of the algorithm's function.
[20 Marks]

Search in the List and Return Relevant Information: The program should be able to search for people by name from the list and return information related to their Coach type and Team name.
You will need to select an appropriate searching algorithm AND provide a clear explanation for why you chose this algorithm and not an alternative.
[15 Marks]

Allow for New User Input (Name, Coach Choice, and Team): The program should be interactive, enabling users to input new data. Users should be able to input a name, choose a Coach type, and select a Team for that person. This data should then be added to the list and appropriately stored in the program's memory. It's essential to validate user inputs to ensure that the Coach type and Team selected actually exist and list all the new entries in the terminal.
[20 Marks]

Generate Random People with Coach Types and Teams: In this step, you will have to create a mechanism for generating random individuals and assigning them a Coach type and Team. This could be used for testing and populating the system with initial data. You will need to ensure that the randomly generated data doesn't conflict with existing Coach types and Teams and list all the players in the terminal.
[30 Marks]

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

Description	Weighting	Mark
Recursive Sorting Algorithm - Implementation & Rationale (i.e. clear reasoning is provided for why this algorithm is chosen)	20	
Searching Algorithm - Implementation & Rationale (i.e. clear reasoning is provided for why this algorithm is chosen)	15	
New User Input (Name, Coach Choice, and Team) & Validation	20	
Generating Random Data [Code must be commented properly!] Use appropriate Data Structure.	30	
Menu utilises enums in a clear & sensible way. The code has been commented properly to explain what has been done	15	

Project Skills and Professionalism

1. **Report (60% weighting)**

You are required to demonstrate your ability to analyse the problems associated with designing and implementing the Rugby Club program. You should research to identify the best practices and solutions in software design and data modelling. You will need to discuss the challenges faced in designing the Rugby Club program and how you formulated solutions to address specific challenges in the assignment. Also, explain how you addressed the specific challenges and give a rationale for the solutions and why they are effective.

In analysing the solution, provide the strengths and weaknesses of your implementation choices. The report should be clear and well-structured, presenting your proposed solutions. Identify any alternative approaches that could have been taken, explain why they were not chosen and discuss the implications of choosing one approach over another. Use a problem definition card that outlines the key challenges and requirements of the Rugby Club assessment.

2. **Problem definition and mapping diagram (10% weighting)**

Additionally, you can create a mapping diagram using draw.io, for example, which will visually represent the relationship and interactions between the different components of the problem in the assignment. A risk assessment and how you chose one solution over another should be discussed based on risk factors.

3. **Presentation (30% weighting)**

Create a three to five-slide PowerPoint and give a five-minute presentation that communicates the task of your assignment and the problem-solving process and solution. The presentation should clearly define the problem—an overview of the chosen solution, including its main components and functionalities. If applicable, show any results or outcomes of implementing the solution. Explain why the selected solution was chosen over alternatives. Highlight the strengths and advantages of the chosen approach. Summarise the key risks identified and the strategies for mitigating them. Present the mapping diagram to illustrate the system's structure and interactions visually. Summarise the key points and emphasise the effectiveness of the proposed solution.

Support your analysis with references and properly reference ALL sources you have used. WARNING – If you do not support your work, you will not receive a high mark!

You should record your presentation, which will be shown in class to your peers, followed by a brief Q&A.

TIMING: You may lose marks if your presentation is shorter or longer than the specified time limits.

NOTE – IF YOU ARE UNABLE TO RECORD YOUR PRESENTATION, THEN YOU WILL BE ALLOWED TO PRESENT 'LIVE' DURING THE CLASS TIME.

Marking Schedule Project Skills and Professionalism:

Description	Weighting
Problem risk assessment. · Identify the main risks of the problem solutions. · Discuss mitigation of risk on solutions taken. · Consider the potential impacts of solutions.	**30%**
Problem solutions were taken and rationale. · The report gives a clear analysis and gives a rationale for the selected solutions.	**30%**
Problem definition and mapping diagram. · Clarify the problem (could use a problem definition card) · Mapping diagram, comprehensively breaking down the problem.	**10%**
Presentation · Presentation is conducted in an articulate manner and clearly conveys findings to peers; Questions are handled well and answered appropriately; Presentation adheres to the time limit set out in the requirements. · Project presentation slides are clear and well set out / well designed.	**30%**
Poor referencing, spelling, grammar and layout will incur marking penalties.	
Total	**100%**

Submission Requirements

All assessment submissions must meet the minimum requirements listed below. Failure to do so may have implications for the mark awarded.

All assessment submissions must:

- Project to be submitted as individual .Java files, report be submitted as a Word document ONLY.
- Maximum word count is 2,000 words.
- Be commented properly to explain your work
- Be submitted by the deadline date specified or be subject to late submission penalties
- Be submitted via Moodle upload
- Use Harvard Referencing when citing third-party material
- Be the student's own work.
- Include the CCT assessment cover page.

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 requested by *contacting your lecturer by email within 1 week of the publication of results.* 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 or access the CCT Learning Space.
- 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.

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

- The use of generative AI tools is **not** permitted in this course for the following activities:
    - Impersonating you in classroom context
    - Completing group work that your group has assigned to you
    - Writing a draft of a writing assignment
    - Writing entire sentences, paragraphs or papers to complete class assignments.

- 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.