DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

University of Strathclyde Science

ADVANCED PROGRAMMING SEMESTER 2 COURSEWORK 2024/25 – BULLS AND COWS PROJECT

Submission	Submission Details	Deadline	Weighting	Marking and Feedback
User Stories	MyPlace	10/2/2025 **9am**	10%	Will take place in labs week commencing 10/2/24, marks returned shortly thereafter
High Level Design (Class Diagram)	MyPlace	24/2/2025 **9am**	10%	Will take place in labs week commencing 24/2/25, marks returned shortly thereafter
Iteration 1	MyPlace	10/3/2025 **9am**	10%	Will take place in labs week commencing 10/3/25, marks returned shortly thereafter
Iteration 2	MyPlace	24/3/2025 **9am**	10%	Will take place in labs week commencing 24/3/25, marks returned shortly thereafter
Iteration 3 (final implementation)	MyPlace	31/3/2025 **9am**	5%	Will take place in labs week commencing 31/3/25, marks returned shortly thereafter
Iteration 3 Final Report	MyPlace	4/4/2025 **9am**	5%	Returned within 3 working weeks

Table 1

Note that the weightings total 50% of your CS207 class mark, reflecting the weighting for this semester.

Failure to comply with the instructions detailed in this specification will result in a minimum 10% mark reduction penalty for the appropriate submission. Late submissions: as this is continuous assessment for which sample solutions will be released shortly after the first two deadlines, it will not be possible to accommodate late submissions after the sample solutions are released. Late submissions prior to sample solutions being released will be penalised at 10% for the first day or part thereof according to University lateness policy and 5% thereafter. Late submissions for the final report will be penalised as per University policy. Where an individual within a team has extenuating circumstances they should contact the lecturer as soon as possible and an appropriate course of action will be determined.

There is a lot of information in this document. Please take the time to read it carefully.

Assessment Overview

The purpose of this assignment is to provide the opportunity to follow the software development lifecycle within a team. This is an important aspect of being a professional in the field as it provides structure and consistency, resulting in projects which are more likely to be successful.

An important challenge here is to explore how the software development lifecycle can be applied in a given context. In the assessment for this semester of Advanced Programming you will be asked to follow a software development lifecycle process in order to develop a Java program to allow users to play bulls and cows. This will be completed in teams of five which can be chosen by students so long as it is within their allocated lab group. No requests to change lab group can be accommodated as personal circumstances will have already been taken into account.

Background

Bulls and Cows is a code breaking game. The aim of the game is to decipher a secret code by trial and error. This code can either be a word or numbers. For the word version of the game the word must be an English word. For the numbers version all numbers must be unique for both the code and each guess. For each guess the number of matches will be given. This will be split into bulls, matches in the right position, and cows, matches in the wrong position. For example:

- Secret number: 1359
- Guess: 1395
- Matches: 2 bulls and 2 cows (The bulls are 1 and 3, the cows are 9 and 5).

See here for more information: https://en.wikipedia.org/wiki/Bulls_and_Cows

Tasks

The aim is to develop software in Java which allows the user to play Bulls and Cows with the computer. The functionality will be similar to https://www.mathsisfun.com/games/bulls-and-cows.html but not identical. The software should also maintain the state of completion of secret codes which the user attempts to decipher, and details about the players including a scoreboard showing player statistics such as number of secret codes successfully deciphered.

The assignment comprises multiple tasks to be completed in teams of 5. The assessment is designed to be completed by a minimum of 4. Due to class size it may be necessary to have some teams of 4. Students will be able to select their own team from those in their lab group. The tasks involved are as follows:

Task 1 – Iteration 0 User Stories

The first task is to identify the requirements of the project through creation of user stories and consideration of acceptance criteria. The submission for this stage is the user stories you write. Whilst you are not expected to write and submit all your acceptance criteria, you should spend time considering at least two appropriate acceptance criteria for each user story as you will be asked about this in the marking lab for this submission. Your submission should contain no more than 14 user stories.

To write the user stories, it will be necessary **first** to **elicit** the **requirements** through conversation with the lecturer who will act as the product owner and customer. Requirements elicitation will take the form of questions posed to a Myplace forum. Each team will be permitted to ask one question but will be also be able to see all other teams questions. The deadline for submitting your question will be at the end of the last lab of the week, i.e., Thursday 13th February at 1pm.

Note you will be best placed to perform well if you commence thinking about the requirements prior to asking a question, as general requirements can be gathered from the details provided in the background. The answers provided serve to provide further clarification where required. Questions similar to "what are the user stories" will not be answered, but will still count as the question for your team.

In addition to the requirements elicitation session, each team will be allocated a teaching assistant or demonstrator who will act as product owners by proxy for the lecturer. Your team will check in with your allocated product owner in each of the labs commencing from week 3 and further clarification can be sought from them. You can of course ask questions of any tutor or demonstrator in the lab who isn't nominally assigned to your team.

Submission of the list of user stories should be to the appropriate slot on MyPlace by the deadline shown in Table 1. The submission should be a single page PDF, and should only be submitted by one member of the team though it is the whole team's responsibility to ensure that the submission made is agreed upon and on time. The submission slot will be configured as a team submission, meaning all team members will be able to see the submission and hence check it.

This submission will be marked and feedback provided in person in the labs, see the dates in Table 1. In the lab you will be asked to show the user stories you submitted on MyPlace, and asked to suggest acceptance criteria for a subset of user stories. The combination of your responses and the submission will be assessed for your mark in this aspect. See the related marking scheme later in this document.

Task 2– Iteration 0 High Level Design: Class Diagram

This task involves creating a **class diagram** which should represent your initial high-level design of the Bulls and Cows game. This should be created based on the initial user stories which will be released after the requirements stage has been completed.

The class diagram should identify the appropriate classes, associations, and an initial attempt at identifying appropriate attributes and operations. It should not include any user interface classes at this stage. You should also include a 1 paragraph rationale for any specific design decisions made (i.e. the "why" of those decisions). This can be included as a note in the diagram or as additional text beneath the diagram.

Submission should be via the appropriate slot on MyPlace by the deadline shown in Table 1. The submission should be a **single page PDF**, and should only be submitted by one member of the team though it is the whole team's responsibility to ensure that the submission made is agreed upon and on time. The submission slot will be configured as a team submission, meaning all team members will be able to see the submission and hence check it.

The class diagram will be marked and feedback provided in person in the labs, see the date in Table 1.

Task 3–Implementation, Testing, and Final Report Iterations 1-3 (2 weeks each)

After class diagrams have been submitted, a sample solution will be released. This **must be used** as the basis for your **implementation**. All .java files, including JUnit java files, for each iteration should be **submitted to MyPlace in addition** to the commits to the **GitLab** working repository. Submission to MyPlace is to ensure a timestamped copy of code at the point of submission for each iteration.

Note: further details of how we will be using GitLab will be released closer to the implementation stage.

Also note that code should not make use of external libraries or frameworks with the exception of JUnit for testing.

Task 3 is split into three iterations of two-week duration.

Iteration 1 and iteration 2 will cover a number of user stories which will be identified from the product backlog by the lecturer (product owner) at the start of each iteration. The third and final iteration will comprise two parts, the first part is the remaining implementation of user stories identified at the start of the final iteration. The second part is a final report, see further details below.

Code for iterations 1, 2, and 3 will be marked in the labs in week 8, 10 and 11 by the tutors and lecturer. See Table 1 for dates.

Workload and Lessons Learned Report

The second part of the final iteration should take place in week 11 and comprises creation of a final workload and lessons learned report which is due on the final Friday of the semester, see Table 1 for the date. The report should take the form of a pdf. The text should be the default font for latex (computer modern) or Word (Calibri light) and should be size 11, which is the default for both LaTex and Word. Equivalent san serif font in size 11 with standard A4 margins is acceptable. You should also include your team name, a title and student team member names. It should be a maximum of **two pages long (plus or minus 1 paragraph, longer results in the application of a non-compliance penalty).**

This report should detail the state of functionality your program achieved, details of the tasks involved in completing the project which team members completed those tasks, a summary of how you worked together as a team, and finally details on lessons learned whilst completing the project and what you would do differently in a team project next time. A suggested Word template will be available on MyPlace.

Peer Assessment

The aggregated and weighted coursework mark for the project may be adjusted proportionately to represent individual contributions to the work. This means that you will be provided "interim" marks at each stage, but these will not be collated, adjusted for individual contribution and finalised until the end of the class.

Adjustments based on individual contribution will be established using a peer evaluation form, review of commits to GitLab, and a workload report submitted by each team. Team members are required to agree a single A4 page workload record and include in the final written report as detailed above.

Team members are required to complete their own personal assessment of their team mates. This will be established using a peer assessment activity on MyPlace. Each member of the team must complete the peer assessment activity on MyPlace by the final deadline detailed in Table 1.

This involves allocating a score of 1-5 to each team member, including yourself, to reflect the individual's contribution to the project. A score of 1 means there was little or no contribution, a score of 5 means they made a significant contribution to the project. If you do not to submit a peer assessment, then it will be assumed all team members made a significant contribution to the project.

Marking Schemes

User Stories

A maximum of 23 marks is available for the initial user stories and acceptance criteria. Marks will be distributed as follows:

Criteria	Maximum Marks Available
Appropriate identification of appropriate role	1
Appropriate identification of an epic user story	1
Appropriate identification of user stories at a suitable level of granularity using correct structure	13
Acceptance criteria questions: appropriate acceptance criteria are presented in the marking lab for the user stories. They follow the suggested structure, demonstrate it shows the user story has been achieved, and are testable. Teams will be asked to identify four acceptance tests, two tests for two user stories which	8

will be randomly selected by the markers. Thus, you should attempt to create at least two acceptance tests per user story you identify.	

Class Diagram

A maximum of 18 marks is available for the high level design. Marks will be distributed as follows:

Criteria	0 points	1 point	2 points	3 points	4 points
Appropriate identification of classes, with clear, narrow responsibilities	No submission	Classes and properties identified are mostly inappropriate for the cryptogram system. Perhaps multiple classes where the responsibilities aren't clear.	Classes and properties appropriately identified are somewhat appropriate, clear areas for improvement such as properties not representing clear responsibilities and unnecessary classes	Classes and properties appropriately identified are mostly appropriate, with perhaps one class which is unnecessary, or some properties do not represent clear responsibilities	Classes and properties identified are appropriate with clear and narrow responsibilities.
Does the design include all the classes needed to provide all the features	No submission	Most features cannot be achieved using this design	Many features can be achieved using this design. This could be e.g. a model class such as something to manage the players is missing	The features can be achieved using this design, perhaps with only a minor issue such as an attribute which is missing	The features can be clearly achieved using this design
Is the design appropriately cohesive	No submission	Some elements of a cohesive overall design, but mostly minimal cohesion. Many instances of attributes and operations in the wrong class or are missing.	Overall good cohesion in some areas, but instances of attributes and operations in the wrong class or are missing.	Overall a design which is very cohesive, but perhaps one or two instances where cohesion could have been improved.	Overall design is highly cohesive with attributes and associations in a single class related to a single purpose.
Is the design appropriately loosely coupled	No submission	Design is very highly coupled	There is some instances of inappropriate coupling	Inappropriate coupling is kept to a minimum	

Syntax is correct	No submission	Syntax is mostly incorrect, with many clear errors	Syntax is mostly correct, with one or two errors such as incorrect cardinality notation	Syntax is correct	

Implementation

Marked in 3 iterations. At the start of each iteration the product owner will select the user stories from the product backlog which should be implemented in that iteration. Each iteration will be marked according to the following criteria and weighted according to the weighting provided in Table 1:

Criterion	0 points	1 point	2 points	3 points
Functionality (per user story)	No functionality achieved	Some functionality of the user story is achieved, but there are significant areas of deviation from the expected functionality as described by the user stories and acceptance tests sample solution	The user story functionality is achieved mostly as expected, perhaps one or two minor cases where the functionality deviates from expected as described by the user stories and acceptance tests sample solution	The user story functionality is achieved completely as expected as described by the user stories and acceptance tests sample solution
Testing (per user story)	No testing	Some attempt at automated testing, but it is significantly limited compared to the criteria prescribed in the sample solution	Testing is mostly as expected, with perhaps one or two scenarios missed	Testing is as expected with all scenarios and acceptance criteria described in the sample solution explored through automated testing.
Code Quality (per iteration)	No submission	Code quality is poor, demonstrated by long unfocussed methods, large code reuse, and inappropriate naming of variables	Code quality is good, but with definite room for improvement. Perhaps a few overly long or unfocussed methods, or some poor variable name choices such as String s1	Code quality is very good, small focussed methods, reuse is apparent, meaningful variable name choice
User Interface (per iteration)	UI is not at all clear, the user wouldn't be able to figure out the	UI is somewhat clear, with a little time the user could figure out the commands but may make one or two mistakes	UI is very clear, the user can easily figure out what the commands are and how to achieve the functionality with no mistakes	

	appropriate commands			

Iteration 3 Final Report

The final report will be marked after submission on MyPlace according to the following criteria.

Criterion	0 points	1 point	2 points	3 points
Team performance	No submission	Little evidence of team collaboration	Satisfactory demonstration of team collaboration, workload is well distributed with perhaps one or two members contributing more	Excellent demonstration of team collaboration, workload is well distributed across all team members
Lessons learned	Submitted artefact demonstrates no reflection on the lessons learned from the project	Submitted artefact demonstrates limited reflection on the lessons learned, perhaps one lesson identified	Submitted artefact demonstrates good reflection on lessons learned with suggested changes for future. However, it may be limited in places.	Submitted artefact demonstrates very good reflection on the lessons learned with clear routes for change in future team projects.
Quality of writing	Writing is unclear and/or disorganised. Thoughts are not expressed in a logical manner. Highly illiterate.	Writing is somewhat unclear and disorganised. Clear areas for improvement. Thoughts are often expressed in illogical manner. Commonly illiterate.	Writing is moderately clear and well organised. Some areas for improvement. Thoughts are mostly expressed in a logical manner. Mostly literate.	Writing is very clear and well organised. One or two small areas for improvement. Thoughts are expressed in a logical manner. Literate, with perhaps one or two minor errors.

Supplementary Material

One possible tool for creating your class diagram is VioletUML, available from http://alexdp.free.fr/violetumleditor/page.php . It is free to download, and can be run from the JAR file thus doesn't need to be installed. You are free to use other software should you so desire.