Cardiff School of Computer Science and Informatics
Coursework Assessment Pro-forma

Module Code:	CMT304
Module Title:	Programming Paradigms
Lecturer:	Víctor Gutiérrez-Basulto and Frank Langbein
Assessment Title:	Logic and functional programming
Assessment Number:	1
Date Set:	28th October 2019
Submission date and Time:	10th December 2019 at 9:30am
Return Date:	14th January 2020

This assignment is worth 25% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;
2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:
`https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf`

Submission Instructions

All submission must be via Learning Central. Upload the following files in a **single zip file**, `[student number].zip`:

Description		Type	Name
Cover Sheet	**Compulsory**	One PDF (.pdf) file	`[student number].pdf`
Task 1	**Compulsory**	One source file	`task1.{lp,hs}` (pick suffix according to programming language chosen)
Task 2	**Compulsory**	One PDF (.pdf) file	`task2.pdf`

Any code submitted will be run on on a system equivalent to those available in the Linux laboratory and must be submitted as stipulated in the instructions above.

Any deviation from the submission instructions above (including the number and types of files submitted) will result in a mark of zero for the assessment or question part.

Staff reserve the right to invite students to a meeting to discuss coursework submissions.

Your submissions will be checked for plagiarism. Your work must be your own and you must independently solve the problem and submit your own solution. Any other

material or sources of information you use must be referenced. Code and text you submit will be compared with other submissions and various other sources on and off the Internet. Any substantial similarities of you submission to unreferenced work or material not created by yourself will be subject to unfair practice procedures. Marks will only be assigned for work you have done yourself (incl. finding and discussing material from references, but not the referenced work; there are no marks for code copied from elsewhere, but for either writing your own code or integrating and adapting code that you have not written).

Assignment

Consider the following situation:

> Patent requests are submitted to the patent office and are reviewed by the technical board. Each patent request is verified and discussed by a group of technical board members chosen by the head of the board. To find a good match between patent request and referees, every member of the technical board submits a bid that classifies all requests that need to be reviewed into four categories depending on their expertise:
>
> EXPERT, KNOWLEDGEABLE, FAMILIAR or INEXPERT

The goal is to write a program to automated this process to help the head of the board. Using a list of bids, indicating the level of expertise for each patent request and board member, it should assign each submitted patent request to a specific number of n members of the technical board such that

- the workloads of the technical board members are approximately equal, that is, do not differ by more than m;
- no member of the technical board is required to review a submission that is placed in the INEXPERT category;
- no member of the technical board is required to review more than k submissions from the FAMILIAR category;
- the total number of cases when a submission is assigned to a member who placed it in the EXPERT category is as large as possible.

The parameters n, m and k are arguments set when calling the program.

Task 1: Write a functional or logic program (in Haskell or ASP) which finds all solutions to the problem, given n, m, k and the list of bids in a suitable format. Make sure you document your code so it is clear how it should be used and what the approach to solving the problem is. Also include your name and student id in the comments. Note, there is no 'correct' programming paradigm to solve this problem. The representation of the problem can be specific to the programming language used.

Task 2: Write a short report on logic vs. functional programming related to the problem:

1. Provide, in up to 300 words, two arguments for and two arguments against using logic programming to solve the problem.

2. Provide, in up to 300 words, two arguments for and two arguments against using functional programming to solve the problem.
3. Justify, in up to 300 words, your choice of programming paradigm, based on your previous arguments.

The word limits are an upper limit, not a target length. Text longer than the word limit for each point will be ignored. Clearly mark each argument in your answer and indicate if it is for and against. Only provide two arguments for and against for the first two points; additional arguments will be ignored.

Learning Outcomes Assessed

Discuss and contrast the issues, features, design and concepts of logic and functional programming.

Criteria for assessment

Task 1: maximum 50 marks, assessed according to the following scale

Fail	0	No code has been submitted.
	$1-14$	Code does not run or does not produce valid output for any valid input; little to no relevant documentation.
	$15-24$	Code is valid without syntax errors and creates a valid output for every valid input (or produces a suitable error message for valid cases it cannot process). Even if the output is not a solution, a suitable attempt to solve the problem is visible. An attempt to document the code has been made.
Pass	$25-29$	Code is valid without syntax errors and creates a valid output for every valid input (or produces a suitable error message for valid cases it cannot process). A suitable attempt to solve the problem has been made, that will often find at least one solution (if there is any). The attempt has been reasonably documented, but no consideration has been given to optimise the program's performance.
Merit	$30-34$	Code is valid without syntax errors and creates a valid output for every valid input (or produces a suitable error message for valid cases it cannot process). A suitable attempt to solve the problem has been made, that will find all solutions (if there are any). The attempt has been well documented.
Distinction	$35-50$	Code is valid without syntax errors and creates a valid output for every valid input. A suitable attempt to solve the problem has been made, that will find all solutions (if there are any) for all problems, with excellent performance. The attempt has been well documented and clearly shows an effort to optimise the program's performance, e.g. by using efficient algorithms and data repres1entations and also some heuristics.

Task 2: maximum 50 marks, assessed according to the following scale

Fail	0	No document has been submitted.
	$1-14$	An insufficient number of arguments has been submitted and/or they hardly apply to the logic and functional programming paradigms. At most an incomplete attempt to justify the choice of paradigm has been made.

	$15 - 24$	An insufficient number of arguments has been submitted, but they show some understanding of the logic and functional programming paradigms. An attempt has been made to justify the choice of paradigm.
Pass	$25 - 29$	The required number of valid arguments has been submitted. They are generally valid for the logic and functional programming pradigm, but they repeat similar issues, do not consider the specific problem or contain mistakes in the details. A suitable attempt has been made to justify the choice of paradigm.
Merit	$30 - 34$	The required number of valid arguments has been submitted. They show a clear understanding of the logic and functional programming paradigms and how these relate to the problem. The choice of paradigm to solve the problem is well justified based on these arguments.
Distinction	$35 - 50$	The required number of valid arguments has been submitted. They show a clear understanding of the logic and functional programming paradigms and the underlying theoretical concepts and/or realisations on programmable machines and how these relate to the problem. The choice of paradigm to solve the problem is well justified based on these arguments and shows an undestanding of related performance issues.

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on 14th January 2020 via Learning Central. This will be supplemented with oral feedback on request.

Feedback from this assignment will be useful for the final exam.