Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CMT 202
Module Title: Distributed and Cloud Computing
Lecturer: Padraig Corcoran
Assessment Title: CMT 202 Coursework 2
Assessment Number: 2
Date Set: Tuesday 10 March 2020.
Submission Date and Time: Monday 27 April 2020 at 9:30am.
Return Date: by Friday 8 May 2020.

This assignment is worth 15 % 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 should be via Learning Central unless agreed in advance with the Director of Teaching.

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Solutions	**Compulsory**	One zip (.zip) file containing the Python code developed.	[student number].zip

Any code submitted will be run on a University provided Linux laptop. The only additional Python library, other than those already installed on the Linux laptop, which will be used to run this code is pyro.

Any deviation from the submission instructions above (including the number and types of files submitted) may 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.

Assignment

You have been hired by a library to build a distributed data storage system using a remote object paradigm that will allow their employees to access, store and update information about books.

Each book has the following seven associated pieces of information which must be stored in the system:
1. Book unique ID (this should be an Integer value in range 1 to n where n is the number of books in the system).
2. Book author.
3. Book ISBN (unique numeric commercial book identifier). This is a sequence of Integer values separated by hyphens e.g. 978-3-16-148410-0.
4. Book title.
5. Book year of publication.
6. Book status (currently *on loan* or *not on loan*; this is a binary indicator).
7. Number of times the book has been loaned. This equals the number of times the status of the book in question has changed from *not on loan* to *on loan*.

Note, multiple copies of the same books can exist in the system, but each individual book should have a unique ID.

Design and implement the above distributed data storage system using a remote object paradigm which allows employees to perform the following seven tasks:

1. Return all seven pieces of information relating to the set of books currently stored in the system. This includes books both on loan and not on loan. Implement this using a method with the following header:
def return_books(self)
The information returned by this method should have the property that it can be easily interpreted by the user when displayed using the Python print function.

2. Add a book to the set of books currently stored; a newly added book should have a status of not on loan. The book unique ID should be returned by the method in question after completion of the task. Implement this using a method with the following header:
def add_book(self, author, title, ISBN, year)
An example of calling this method is:
library_object.add_book('Allen Hatcher', 'Algebraic Topology', '978-05210-79-5401', 2019)
The information returned by this method should have the property that it can be easily interpreted by the user when displayed using the Python print function.

3. Return all information relating to the subset of books currently stored which have a publication date within a specified inclusive year range (e.g. 2000-2010). Implement this using a method with the following header:
def select_by_year(self, start_year, end_year)
An example of calling this method is:
library_object. select_by_year(2000, 2020)

The information returned by this method should have the property that it can be easily interpreted by the user when displayed using the Python print function.

4. Set the status of all books with a specified title to on loan. Implement this using a method with the following header:
def set_on_loan(self, title)
An example of calling this method is:
library_object.set_on_loan('Algebraic Topology')

5. Set the status of all books with a specified title to not on loan. Implement this using a method with the following header:
def set_not_loan(self, title)
An example of calling this method is:
library_object.set_not_loan('Algebraic Topology')

6. Return all information relating to the set of books currently stored sorted in descending order by year of publication; that is, most recently published first. Implement this using a method with the following header:
def return_books_sorted(self)
The information returned by this method should have the property that it can be easily interpreted by the user when displayed using the Python print function.

7. Return all information relating to the subset of books currently stored where the corresponding ISBN contains a specified Integer value. For example, the ISBN 978-3-16-148410-0 contains the Integer 16 but does not contain the Integer 6. Implement this using a method with the following header:
def return_books_ISBN(self, intValue)
An example of calling this method is:
library_object. return_books_ISBN(16)
The information returned by this method should have the property that it can be interpreted by the user when displayed using the Python print function.

The class in question should be called Library; that is, when defining the class use the expression "class Library(object):". Also, the class must be contained in a file entitled Library.py.

Note that, to successfully complete all tasks, you are only required to stored data while you code is running; there is no requirement to write data to an external database or file.

Learning Outcomes Assessed

The following learning outcomes from the module description are specifically being assessed in this assignment.
Demonstrate and apply knowledge about the state-of-the-art in distributed-systems architectures.
Appreciate the difference between various distributed computing middleware and their communication mechanisms.

Criteria for assessment

Credit will be awarded against the following criteria.

Marks will be assigned based on successful system implementation as follows:
Successfully implement task 1 specified above. [2 marks]
Successfully implement task 2 specified above. [2 marks]
Successfully implement task 3 specified above. [2 marks]
Successfully implement task 4 specified above. [2 marks]
Successfully implement task 5 specified above. [2 marks]
Successfully implement task 6 specified above. [1 mark]
Successfully implement task 7 specified above. [1 mark]
The number of times a book has been loaned is correctly calculated. [1 mark]
Ensure book unique ID is an Integer value in range 1 to n where n is the number of books in the system. [1 mark]
Using a name server. [1 mark]

Feedback on your performance will address each of these criteria.

A student can expect to receive a distinction (70-100%) if they correctly implement all tasks.
A student can expect to receive a merit (60-69%) if they correctly implement most tasks without major errors.
A student can expect to receive a pass (50-59%) if they correctly implement some tasks without major errors.
A student can expect to receive a fail (0-50%) if they fail to correctly implement some tasks without major errors.

IMPORTANT – All code submitted must be written in Python 3 and use the pyro library to implement remote objects. Any code submitted will be run on a University provided Linux laptop. The only additional Python library, other than those already installed on the Linux laptop, which will be used to run this code is pyro.

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on Friday 8 May 2020 via Learning Central. Where requested, this will be supplemented with oral feedback.