Assignment paper

Faculty	Information Technology		
Qualification	Bachelor of Science in Information Technology		
Module code	ITJA321	**Module name**	Java and Distributed Systems
Module leader	Mr. Dennis Luke Owuor	**Internal moderator**	Ndai Makhurane
Copy-editor	Mrs Carol Fofo	**Total marks**	158
Assignment title	Assignment 1	**Due date**	12 October 2018

This module is presented on NQF level 7.
Mark deduction of 5% per day will be applied to late submission.
Late assignments will not be accepted after three days from due date.
This is an individual assignment.

This assignment contributes 20% towards the final mark.

Instructions to candidate

- Remember to keep a copy of all submitted assignments.
- All work must be typed.
- Please note that you will be evaluated on your writing skills in all your assignments.
- All work must be submitted through Turnitin[1] and the full Originality Report must be submitted with the final assignment.
- Negative marking will be applied if you are found guilty of plagiarism, poor writing skills or if you have applied incorrect or insufficient referencing. (See the table at the end of this document where the application of negative marking is explained.)
- Each assignment must include a cover page, table of contents and full bibliography, based on the Harvard Referencing Style as applied at CTI Education Group.
- Use the cover sheet template[2] for the assignment. This is available from your lecturer.
- Students are **not** allowed to offer their work for sale or to purchase the work of other students. This includes the use of professional assignment writers and websites, such as Essay Box. If this should happen, CTI Education Group reserves the right **not** to accept future submissions from a student.

Assignment format

Students must follow the generic requirements when writing and submitting assignments as follows:

- Use standard Arial, font size 10.
- Include page numbers.
- Include a title page.
- Print submissions on both sides of the page.
- Write no more than the maximum word limit.
- Ensure any diagrams, screen shots and PowerPoint presentations fit correctly on the page and are referenced.
- Include a table of contents.
- Use accurate Harvard referencing throughout the assignment.
- Include a bibliography based on Harvard Referencing System at the end of the assignment.

[1] Refer to the **CTI Plagiarism Policy**, which is available from your lecturer.

[2] Available on myLMS.

- Include completed Assignment Front Cover Sheet and Statement and Confirmation of Own Work (available on *my*LMS).
- Check spelling, grammar and punctuation.
- Run the assignment through Turnitin software.
- Students must keep copies of all submitted work.

Essential embedded knowledge and skills required of students
- Report-writing skills
- Ability to analyse scenarios/case studies
- Understanding of subject field concepts and definitions
- Ability to apply theoretical knowledge to propose solutions to real-world problems
- Referencing skills (Harvard Referencing Method)

Resource requirements
- A device with Internet access for research
- A desktop or PC for typing assignments
- Access to a library or resource centre
- Prescribed reading resources

Delivery requirements (evidence to be presented by students)

- A typed assignment[3]
- A Turnitin Originality Report

Minimum reference requirements
At least five references for first year, ten references for second year and fifteen references for third year.

Additional reading is required to complete this assignment successfully. You need to include the following additional information sources:

- Printed textbooks/e-books
- Printed/online journal articles
- Periodical articles (e.g. business magazine articles)
- Information or articles from relevant websites
- Other information sources, e.g. geographic information (maps), census reports, interviews, etc.

Note
• It is crucial that students reference **all** consulted information sources, by means of in-text referencing and a bibliography, according to the Harvard referencing style. • Negative marking will be applied if a student commits plagiarism (i.e. using information from information sources without acknowledgement and reference to the original source). • In such cases, **negative marking**, also known as '**penalty scoring**', refers to the practice of subtracting marks for insufficient/incorrect referencing. • Consult the table at the end of this document, which outlines how negative marking will be applied as well as the way in which it will affect your assignment mark.

[3] Refer to the **CTI Conditions of Enrolment** for more guidance (available on myLMS).

Assessment criteria

The following criteria are assessed in this assignment:

LO1	**Comply with the goals, architecture and changes of distributed systems.**	**Question no.**	**Textbook section**
1.1	Understand the concept of 'distributed computing.	1.1	Appendix A
LO2	**Apply the concept of 'parameter passing' of Java objects.**	**Question no.**	**Textbook section**
2.2	Use selections, loops and methods.	1.3, 1.8	Pp 179
2.3	Define classes and objects.	1.4, 1.6, 1.7, 1.8	Pp 343
2.4	Understand inheritance and polymorphism.	1.5, 1.6, 1.7, 1.8	Pp 431
LO3	**Explore the relationships among Java Collection Framework and the ability to traverse a Collection.**	**Question no.**	**Textbook section**
3.1	Traverse and create instances of the Java Collection Framework interface.	1.8	Pp 784
3.2	Differentiate between HashSet, LinkedSet and TreeSet.	1.10	Pp 819
3.3	Know when to use ArrayList rather than LinkedList.	1.8	Pp 783
LO6	**Create a simple database application in Java using JDBC.**	**Question no.**	**Textbook section**
6.1	Load database drivers and execute SQL statements.	1.2, 1.9	Pp 1200, 1211
6.2	Use the PrepareStatement.	1.2, 1.9	Pp 1219

Question 1 — 45 Marks

1.1 Create two classes in the same package. Call the first class OverloadDemo.java, while the second class call it Mainclass.java. The OverloadDemo class overload method Area(), to be able to calculate area of Square, Rectangle and Circle. Implement the method area() on the Mainclass.java. (15 marks)

1.2 As one of the outstanding Java programmer at Pearson Institute of Higher Learning, the department of IT has requested you to give a one hour lecture with details code example to the second year IT students with regard to the topics below:

a. With the aid of flow chat diagrams and code examples, explain in detail the difference between **Do..while loop** and **while loop**
(15 marks)

b. Give a detailed explanation of the difference between Implement Runnable vs Extend Thread in Java. (15 marks)

End of Question 1

Question 2 — 113 Marks

[E-attendance register]

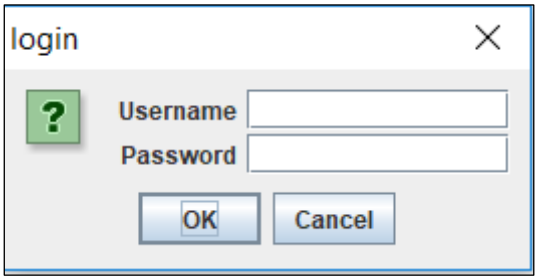
Pearson Institute of higher learning has embarked on a program to reduce paper consumption across its university in South Africa. The goal is to reduce paper consumption by 40% by the beginning of July 2019. The faculty of IT has decided that in order to help the institution achieve this goal, they want to reduce printings by doing away with paper-based class attendance registers. As a final year IT student, the faculty has requested you to help them come up with an e-attendance register. The faculty has presented to you the following system specification to guide you on the development.

The system specifications:

- To achieve the above goal, choose a web server solution of your choice, I,e xampp or wampp.
- Create your own database, create two tables, i.e. login and register tables.
- On the login table which has two columns (username and password) populate with authorised usernames and passwords.
- Build your java-based desktop application.
- The application first requests the lecturer to login. This is achieved with the aid of multi-input dialogue box with two input areas for the lecturer to enter the username and password.
- The application then establishes communication with the webserver and confirms the username and password in the login table.
- Once login details have been validated, either a message box displaying, successful with ok button should be displayed, or one with a message, unsuccessful is displayed.
- If successful, the system then displays an input dialogue box for the lecturer to input total number of student in his class.
- Then the system display a multi-input dialogue for each student to input his details, i.e Name, Surname, Student number and Subject until the total number entered matches the number that the lecturer requested.
- These details are stored on the Class-register Table in the database.
- In order to help the lecturer validate this information instantly, the system should be able to store the information on a list-Array and display the entire list-Array data on the console.
- Finally, information displayed on the console should not have duplicate information. Hint: you might use the student number as a unique identifier to help identify duplicated information.

Source: Owuor (2018)

2.1 Create a project in Java, and in the project create a class called login. Inside that class create a method called myLogin() that will be able to create gui for login as shown below. (10 marks)



2.2 On the same package as login, create a Plan Old Java Object (POJO) class called Student, declare this class String name, String surName, int StudentNo and String subject, then create the getters and setters methods for these variable. (20 marks)

2.3 Create another class called registerClass. In this class create a method called private **boolean validate_login(**String username,String password**)**, this method should be able to establish connection to the xampp/wampp database and validate if the username and password entered in Question 1.2 above is correct. (10 marks)

2.4 If the login is successful, the system displays input dialog to allow the lecturer to enter the total number of students in his class. This should then allow the system to determine how many times it will loop students details input dialog box. All the above process should be put in a method called **studentpresent().** It should be called on the main method only, and only if the username and password are correct. These details entered should be temporarily stored in a string variable, except the Studentno, which is an integer. (25 marks)

2.5 Declare **insertDetails()** method to allow all the student details entered to be stored in the database. Make sure that the system automatically inserts today's date. This method should be looped until the entire students' records are in the database. (20 marks)

2.6 Finally for the lecturer to instantly verify the information entered, the system should display all the data stored in the ListArray, which should be similar information with the database information. For efficiency and effectiveness, the arraylist should not have duplicate data. Data will only be duplicated if the information entered share the same student number. (20 marks)

End of Question 2

You have now reached the end of this assignment. Ensure that you have answered all the required questions before submitting your assignment to your lecturer and ensure that you have adhered to all the instructions within this assignment.

Negative marking

Third-year students

- A minimum of 15 additional information sources must be consulted and correctly cited.
- If no additional information sources have been used, a full 15% must be deducted.
- Deduct 1% per missing resource of the required 15. For example:
  - If only five resources cited, deduct 10%
  - If only three resources cited, deduct 12%
- Markers must interpret the Turnitin report to determine actual Overall Similarity Index percentage.
- Markers to apply the penalties for Category A for insufficient sources and incorrect referencing style.
- Markers to apply the penalties/actions for Category B for plagiarism.

Category A

Minimum reference requirements	Deduction of final mark
No additional information sources have been used or referenced	15%

Category B: Interpretation of Turnitin report

Students may not have more than a 15% Overall Similarity Index on Turnitin, after analysis of the report.

Interpretation of Turnitin Originality Report	
Lecturer to capture the following	
1.	Original Overall Similarity Index (percentage) of Turnitin report
2.	Overall Similarity Index (percentage) after lecturer analysis of Turnitin report (to determine legitimate plagiarism)

Penalties		Action
a.	Less than 15% of the body of assessment (based on Point 2 above)	No action. Mark according to memorandum
b.	More than 15% of the body of assessment and first offence (based on Point 2 above)	Award 0% for the assignment
c.	If more than 70% of the body of assessment	Award 0% and conduct disciplinary hearing