NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Faculty of Computing and Informatics

Department of Computer Sciences

13 Storch Street
Private Bag 13388
Windhoek
NAMIBIA

T: +264 61 207 2258
F: +264 61 207 9258
E: csi@nust.na
W: www.nust.na

# COURSE ASSESSMENT 2022

**COURSE DETAILS**

| | | | |
|---|---|---|---|
| Course Code: | PRG621S | Semester: | 1 |
| Course Title: | Programming 2 | | |
| Lecturers: | Co-ordinator: | Mr. H. Kandjimi | |
| | Dr. A. Azeta, Ms N. Nashandi | | |

**COURSEWORK DETAILS**

| | | | |
|---|---|---|---|
| Assessment Number: | 4 | of | 4 |
| Title of Assessment: | Building a real life application based on OOP concepts in Java | | |
| Formats: | Demonstration | Source Code (Program) | |
| Method of Working: | Group/Team work | | |
| Workload Guidance: | Expect to spend about 35 hours on this assessment | | |

**PUBLICATION**

| | |
|---|---|
| Date of Issue: | Wednesday 11 May, 2022 (Semester 1) |

**SUBMISSION**

| | |
|---|---|
| Soft Copy: | All Submission shall be done on e-Learning. Submit your work as a zipped folder named Group_Assignment01 (i.e. Group_Assignment01.zip) BEFORE 23h59 on Wednesday, 11 May 2022. NB: INCLUDE A TEXT DOCUMENT WITH A LIST OF ALL TEAM MEMBERS IN YOUR SUBMISSION. |
| Hard Copy: | Not required (If required will be communicated) |
| Late Submission: | Late submission will not be entertained without prior approval |

| | | | | |
|---|---|---|---|---|
| Multiple Hand-ins: | Multiple hand-ins before the due date is allowed but the latest version will be graded. | | | |
| Time and Date for Submission Closing: | **Date** | Wednesday, 11 May 2022 | **Time** | 23h59 |

**ASSESSMENT**

| The Assessment is: | 30% | out of | 100% final mark |
|---|---|---|---|

**Assessment Strategy**

The assessment strategy is designed to evaluate the student's achievement of the course learning outcomes, and is subdivided as follows:

| Learning Outcome ID | Description | Method of Assessment |
|---|---|---|
| 1 | - Demonstrate an understanding of the core concepts in object-oriented programming paradigm | Demonstration |
| 2 | - Apply object-oriented concepts in application analysis, de-sign, implementation and testing.<br>- Apply generics programming in classes data structures and methods;<br>- Carry-out exceptions and apply object serialization in data structures<br>- Apply good programming style and demonstrate an under-standing of the impact of style on developing and maintaining programs | Source code |
| 3 | - Produce well-documented and elegant programs written in Java and use the object-oriented technique to analyse software problems<br>- Examine the efficiency and testing in software development | Documentation and Testing |
| 4 | - Set up virtual machines and deploy programs in containers | Deployment |

**FEEDBACK**

| | |
|---|---|
| Feedback will be given via: | E-Learning |
| How long after submission will it be available: | Not later than 2 weeks |

## ASSIGNMENT DETAILS

The main aim of this assignment is for the students produce well-documented and elegant programs written in Java with the use of object-oriented technique to analyse software problems and be able to implement most if not all concepts learnt in theory. For the sake of this assignment we will consider a Library management system.

A typical Library management system has multiple objects interacting and interrelated within the system to allow for most operations to be caried out. Sample objects to be considered are general people using the system that could further be categorized as users or authors and additionally be implemented as students or staff members. Further objects within the library could be books and other equipment or materials within the system.

The system should have the basic operations such as authentication with different user roles and management of users. Different users should be able to borrow books or request usage of any facility withing the library which should additionally be tracked by time to indicate return time or check out time in cases of facility usage.
To allow for creativity and diversified thinking, the assignment description above is a basic skeleton of the requirement and hence it's up to the groups to explore and come up with their own system requirements.

Below are the steps that may help you kick-start the assignment:

1. Read widely on Library management systems and share ideas with group members. Take note of 5 marks for being exceptional in the design, that is, going beyond given requirements;
2. Identify the main objects and operations you need to fulfil requirements.
3. Choose a design pattern that best suits the context and justify your choice. In some cases, you may have to combine design patterns to answer requirements;
4. Draw the UML class diagram in the given context before embarking on code.
5. Explore versioning and collaboration tools to use throughout the assignment

Collaboration and Communication Tools

Below are some of the tools used in agile software development. Some have overlapping functions:

- Communication, e.g., Ms Teams ,WhatsApp,
- Project management, e.g., Trello, Liquid Planner, Jira
- Storing code and version controlling, e.g., Gitlab, Github, sourceForge etc.

**PART I : Program Functionality**

**The program / Application approach is entirely up to you, so long as it follows the description above. In addition innovation and creativity will be an added advantage, however below are guidelines to follow:**

*Section A: Planning and Design*

| **Criteria / Assessment** | **Poor** | **Below Expectation** | **Average** | **Moderate** | **Above Expectation** |
|---|---|---|---|---|---|
| **Justification of Design Pattern** | No relation with the problem (0) | Slight related to problem (3) | Related to the problem (6) | Closely linked to the problem (8) | Suits problem perfectly (10) |
| **UML Class Diagram** | No relation with choice (0) | Relation not well explained (2) | Slight relation with choice(4) | Closely related to choice (7) | Matches choice (10) |
| **Plagiarism Report/Check** | >40 (-15) | <40% (-7) | <30% (3) | <20% (4) | < 10% (5) |

*Section B: Development or Implementation*

| Criteria / Score | Poor | Below Expectation | Average | Expectated | Exceptional |
|---|---|---|---|---|---|
| **Objects** | Object identified have incomplete classes (1) | Only three objects identified (3) | Missing / combined objects (8) | Sound objects, but classes not standard (11) | Objects suits requirements (15) |
| **OOP Concepts** | Zero demonstrated (0) | At least one demonstrated (3) | At least two concepts shown (8) | Some suitable concepts missing (10) | All appropriate concepts shown (12) |
| **Exceptions and Usability, etc.** | Poor usability always crushing (1) | Basic usability but a lot of errors(3) | Some moderate usage (5) | Most exceptions handled (7) | Sound usability with all appropriate errors handle (10) |
| **Communication and collaboration** | No evidence of collaboration(0) | Only communication or collaboration tool used(1) | communication and collaboration tools used (2) | Extensive use of tools throughout(3) | Clear outline collaboration and communication with recommended tools(8) |
| **Data Storage** | No storage used(0) | Basic use of static data structure(1) | Uses text files (2) | More organized data/text files(3) | Connect to database for data manipulations(5) |
| **Sound coding practices** | No style considerations (0) | Inappropriate indentations (1) | Adequate and moderate style(2) | Code easy to read and navigate(3) | Well styled and follows programming conventions (5) |

**PART II: Program formatting and presentation**

**The source code will be marked according to the following indicators.**

1. Good modular designs within class files, different methods per functionality **[03 marks]**
2. Good comments **[03 marks]**
3. Ability to explain a portion of the code as may be required by the evaluator **[15 marks]**

**Part III: Further information**

4. NB: NO PRESENTATION NO MARKS
5. EXTRA MARKS WILL BE AWARDED FOR CRITICAL THINKING, CREATIVE IDEAS AND MOST PRACTICAL APPLICATION
6. PLEASE NOTE THAT INABILITY TO EXPLAIN YOUR CODE WILL LEAD TO ALL THE MARKS BEING PEGGED AT ZERO(SEE TEACHING PHYLOSOPHY IN COURSE OUTLINE).
7. PLAIGIRISM WILL NOT BE TOLORATED AND HENCE IF ASSIGNMENTS ARE COPIED OR SHARED, THEN ALL STUDENTS INVOLVED ARE DISQUALIFIED(GRADED ZERO).

**Total Marks: 100**