BUCKINGHAMSHIRE NEW UNIVERSITY
EST. 1891

Assignment Brief **Academic Year 2024-25**

Module code and title:	COM5043 Object Oriented Programming (24WINDL)	**Module leader:**	Justin Cross
Assignment No. and type:	CW1 Individual Project	**Assessment weighting:**	100%
Submission time and date:	**Mon 2nd June 2025 at 2pm**	**Target feedback time and date:**	23rd June 2025 *(3 weeks after submission)*

Assignment Brief

SCENARIO: Warehouse Management System for BNU Industry Solutions Ltd.

Company Background: BNU Industry Solutions Ltd. is a leading provider of industrial equipment and supplies, serving clients across various sectors including manufacturing, construction, and logistics. As the business grows, the company faces challenges in managing its warehouse operations efficiently.

Current Challenges:

1. **Inventory Tracking:** The company struggles with accurately tracking stock levels, leading to overstocking or stockouts.
2. **Order Fulfilment:** Processing orders manually is time-consuming and prone to errors, affecting customer satisfaction.
3. **Supplier Coordination:** Managing supplier information and orders is cumbersome, resulting in delays and miscommunications.

Objective: To address these challenges, BNU Industry Solutions Ltd. decides to implement a new Warehouse Management System (WMS) that will streamline operations, improve accuracy, and enhance overall efficiency.

Key Features of the Warehouse Management System:

1. **Supplier Management:**
   - Maintain basic supplier information, including contact details and order history.
   - Implement features to add, update, and delete supplier records.
   - Create purchase orders for new inventory, track order status, and manage deliveries.
2. **Inventory Management:**
   - Track and manage stock levels.
   - Receive new inventory and update stock quantities.
   - Monitor low stock alerts to prevent stockouts.
3. **Order Processing:**
   - Process customer orders, and update inventory levels accordingly.
4. **Simple Financial Accounts:**
   - Track financial transactions related to inventory purchases and sales.
   - Generate basic financial reports such as sales summaries and expense reports.

Design, Implementation and Testing (Individual Work - 100 marks)

Based on the scenario above, design and implement the system using object-oriented techniques using an appropriate object-oriented language (e.g. Java, C#, C++, Python, TypeScript).

> On the expectation that the system will grow into a larger and more complex system, use the chosen language's type system, or type annotations, to minimise the likelihood of runtime errors.

Your system will require a user interface to drive the system. This can be either a text based / menu driven interface or a graphical interface. Your system should be structured such that the user interface is decoupled from the business logic.

> **Note:** The assessment will focus on functional behaviour and program structure, rather that the frontend user interface. A text-based interface may be easier to implement than an event driven / graphical one.

The following table shows how the marks are allocated for different tasks:

Stage	Percentage	Deliverable	Description	Mark Available
Design	20%	UML / formal diagrams	This should include suitable UML diagrams to adequately represent the system. This should include a class diagram as a minimum. Other diagram types may also be helpful. Indicate multiplicity, aggregations, and compositions as applicable.	10
		Detailed Design and Rationale	There should be a clear indication of how you have exploited the OO paradigm, with respect to concepts such as Inheritance, Encapsulation and Polymorphism. Include your design of the User Interface.	10
Implementation	50%	Inventory Management	Enable the system to track and report on the stock levels of different products, alerting users to low stock levels.	10
		Supplier Management	Enable the system to manage suppliers, process purchase orders, and receive deliveries.	20
		Order Processing	Enable the system to process customer orders. (For simplicity, only accept orders for available stock, and assume immediate payment and shipment.)	10
		Financial reporting	Provide a financial report showing: • Stock purchase costs. • Sales revenue. • Net Income, indicating profit or loss.	10
Testing & Reflection	30%	Quality Assurance / Testing	Establish an appropriate Test Strategy highlighting how all the stated requirements for your system have been addressed. Include automated tests for most, if not all, aspects of the system.	20
		Reflective Report	A reflective report, including an explanation of the process used to produce the above models/diagrams stating any constraints and assumptions used along with any difficulties encountered and the course of action taken to overcome them. Also include what you have learned during the implementation and what you might choose to do differently next time.	10
		Total		**100**

You should use GitHub throughout the development process, with regular commits, to show how the system has been developed over time. Include the GitHub link to your source code within the submitted report. Also copy and paste your source code of the final system as text as an appendix into the report.

This assignment has been designed to provide you with an opportunity to demonstrate your achievement of the following module learning outcomes:
LO1. Apply the fundamental concepts and principles of the object-oriented approach to the complete software lifecycle, but with particular emphasis on the backend.
LO2. Analyse a given object-oriented design in terms of the required implementation in a designated high-level object-oriented programming language.
LO3. Implement a solution using an object-oriented methodology, with all the required supporting documentation and where appropriate illustrating features of inheritance, encapsulation, and polymorphism, within the OO programming language you choose.
LO4. Utilise CASE development tools for the analysis and solving of given problem scenarios.

Referencing and research requirements
Please reference your work according to the Harvard style as defined in *Cite Them Right Online* (http://www.citethemrightonline.com). This information is also available in book form: Pears, R. and Shields, G. (2019) *Cite them right: the essential reference guide.* 11th edn. Basingstoke: Palgrave Macmillan. Copies are available via the University library.

How your work will be assessed
Your work will be assessed against the assessment criteria which have been provided at the end of this brief. These criteria have been designed specifically for this assignment and are intended to measure the extent to which you have demonstrated your achievement of its associated learning outcomes (see above). They have been aligned with the institutional grade descriptor appropriate for your level. The assessment criteria provide a basis for fair and consistent marking and indicate what is expected of you in this assignment. It is strongly recommended that you engage with them while you are working on the assignment and use them in combination with any feedback you receive once your work has been marked to help you plan for future learning and development.

Submission details
• **Your assignment should be submitted by 14:00pm UK time on the specified day (see Submission date and time, below) via the TurnItIn Submission portal. Please do not submit assignments directly to lecturers or other university staff.** • Please retain a copy of your assignment for reference, as you may be required to produce it in the future. • You are reminded of the University's regulations on academic integrity, which can be viewed on the University website: https://www.bucks.ac.uk/sites/default/files/2021-07/academic-integrity-policy.pdf. In submitting your assignment, you are acknowledging that you have read and understood these regulations. • Please also note that work that is submitted up to 10 working days beyond the submission date will be considered a late submission. Late submissions will be marked and the actual mark recorded, but will be capped at the pass mark (typically 40%), provided that the work is of a passing standard. Work submitted after this period will not be marked and will be treated as a non-submission.

Level 5	**Fail**		**Pass**				
Grade Band	**(1,10,20) (F) Fail** Not successful	**(32,35,38) (E) Marginal Fail** Below required standard	**(42,45,48) (D) Satisfactory**	**(52,55,58) (C) Good**	**(62,65,68) (B) Very Good**	**(72,75,78) (A) Excellent**	**(82,85,88,92,95,98,100) (A+) Outstanding**
Grading Categories	Not successful	Below Required Standard	Satisfactory	Good	Very good	Excellent	Outstanding
Criterion 1 (LO1) Apply the fundamental concepts and principles of the object-oriented approach to the software lifecycle	No understanding of object-oriented concepts or lifecycle application.	Limited understanding, with irrelevant or minimal application.	Basic understanding with some relevance to the software lifecycle.	Good understanding and application of object-oriented concepts relevant to the lifecycle.	Very good application of object-oriented principles with clear lifecycle relevance.	Excellent application, with insightful and precise understanding of object-oriented principles in the lifecycle.	Outstanding, innovative application demonstrating advanced understanding of object-oriented principles and their lifecycle relevance.
Criterion 2 (LO2) Investigate and interpret object-oriented designs for implementation in an OO programming language	No investigation or interpretation of designs. No evidence of OO programming understanding.	Minimal investigation or incorrect interpretation. Limited programming application.	Basic investigation and interpretation with limited programming functionality.	Good investigation and interpretation, leading to functional implementation in an OO programming language.	Very good investigation and effective implementation using OO programming.	Excellent investigation, with advanced and efficient programming implementation.	Exceptional investigation and innovative implementation, showing mastery of OO programming concepts.

Internal approval: Jon Jackson, 28/11/2022

Criterion 3 (LO3) Implement a solution using an object-oriented methodology with documentation illustrating inheritance, encapsulation, and polymorphism	No implementation or relevance to object-oriented principles. Documentation missing or incomplete.	Minimal implementation with limited use of inheritance, encapsulation, or polymorphism. Poor documentation.	Basic implementation showing some use of object-oriented principles. Documentation is adequate.	Functional implementation with clear use of object-oriented principles including **inheritance**, **encapsulation**, and **polymorphism**. Documentation is clear and sufficient.	Effective implementation demonstrating very good use of object-oriented principles including inheritance, encapsulation, and polymorphism. Documentation is **thorough**.	Advanced implementation with excellent use of object-oriented principles including inheritance, encapsulation, and polymorphism. Documentation is **professional and detailed**.	Exceptional implementation showcasing mastery of object-oriented principles including inheritance, encapsulation, and polymorphism. Documentation is **comprehensive and innovative**.
Criterion 4 (LO4) Utilise CASE development tools for the analysis and solving of given problem scenarios	No use of CASE tools or evidence of their application in problem-solving	Minimal or incorrect use of CASE tools, showing limited understanding.	Basic use of CASE tools to analyse and address problem scenarios.	Good use of CASE tools, effectively applying them to analyse and solve problems.	Very good use of CASE tools, demonstrating solid problem analysis and solution development.	Excellent use of CASE tools, showing advanced application in problem-solving and analysis.	Exceptional use of CASE tools with innovative and comprehensive solutions to problem scenarios