1st SIT Independent Learning Question Paper   **Year Long 2024 - 2025**

Module Code:	**CS4001NT**
Module Title:	**Programming**
Module Leader:	**Mr. Jeevan Poudel / Mr. Mohit Sharma (Islington College)**

Coursework Type:	Individual
Coursework Weight:	This assessment accounts for *10%* of the overall module grades.
Submission Date:	Friday, 2 May 2025
Coursework given out:	Week 24
Submission Instructions:	Submit the following to the Itahari International College's MST portal before 01:00 PM on the due date: ● A report (document) in .pdf format in the MST portal or through any medium which the module leader specifies.
Warning:	London Metropolitan University and Itahari International College takes plagiarism very seriously. Offenders will be dealt with sternly.

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed overleaf. Please sign below to say that you have read and understand these extracts:

Extracts from University *Regulations on Cheating, Plagiarism and Collusion*

Section 2.3: *"The following broad types of offence can be identified and are provided as indicative examples ….*

(i)   Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii)   Falsifying data in experimental results.

(iii)   Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv)   Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v)   Collusion to present joint work as the work solely of one individual.

(vi)   Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1.   Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation, and computer programs.

2.   Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "e = mc^2 (Einstein 1905)". A *reference* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

School of Computing, FLSC

CONTRACT CHEATING

Contract cheating (also known as assessment outsourcing, commissioning or ghost writing) is when someone seeks out another party, or AI generator service, to produce work or buy an essay or assignment, either already written or specifically written for them or the assignment to submit as their own piece of work.

Contract cheating undermines the integrity of the academic process and devalues the qualifications awarded by the university. Students are reminded that academic integrity is a fundamental principle of our institution. Engaging in contract cheating not only impacts the individual's academic record but also the reputation of the university.

Students are encouraged to seek support if they are struggling with their coursework. The university offers a range of resources, including academic counseling, tutoring services, and workshops on study skills and time management. Utilizing these resources can help students achieve their academic goals without resorting to dishonest practices.

Penalty:

- Failure in the Module: The student must re-register for the same module, and the re-registered module will be capped at a bare pass.

- Ineligibility to Continue on the Course: Where re-registration of the same module, or a suitable alternative, is not permissible, the student will not be able to continue on the course. Additionally, the following penalty will be applied to the student's final award:

   - Undergraduate Honors: The student's final classification will be reduced by one level.
   - Unclassified Bachelors: Downgraded to Diploma in Higher Education.
   - Foundation Degree: Distinction downgraded to Merit; Merit downgraded to Pass; Pass downgraded to Certificate in Higher Education.
   - Masters: Distinction downgraded to Merit; Merit downgraded to Pass; Pass downgraded to Postgraduate Diploma.

Reporting and Consequences:

Instances of contract cheating will be thoroughly investigated, and students found guilty will face the penalties outlined above. It is the responsibility of every student to ensure that their work is their own and to avoid situations that could lead to accusations of academic misconduct.

By adhering to these standards, students contribute to a fair and equitable academic environment, ensuring the value and recognition of their qualifications are maintained.

Assessment

This assignment will be marked out of 100 and carries 10% of the overall module weighting.
Your report for this part must be uploaded and submitted by RTE Deadline. The assignment must be carried out individually so you must not obtain help from anyone other than the module teaching staff. Collusion, plagiarism (unreferenced copying), and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension.

Aim

The aim of this assignment is to enable students to demonstrate their understanding and application of object-oriented programming (OOP) concepts using Java. Through this technical report, students will apply core OOP principles, develop problem-solving skills, and enhance their ability to communicate technical concepts. The assignment encourages independent learning and reflection on their understanding of Java, preparing them for real-world scenarios and professional development in software engineering.

Deliverables

Students will be required to **complete** the assigned **LinkedIn Learning course on Java Object-Oriented Programming** and submit a comprehensive technical report that answers specific questions related to the course content. The report should demonstrate **their understanding of OOP principles**, including encapsulation, inheritance, polymorphism, and abstraction, with relevant Java code examples to illustrate these concepts. Additionally, students should provide **evidence** of course completion, such as a **screenshot or completion certificate**. All materials should be compiled into a single pdf document and submitted via MST by the specified deadline.

Technical Report Questions

Student must provide evidence [appropriate screenshots of programs (in blueJ) or figures] that is carried out while documenting the report.

1. a. What are the four main principles of Object-Oriented Programming(OOP)? Provide a brief explanation of each.

   (5 marks)

   b. How does the concept of a class differ from that of an object in Java? Provide examples from the course to illustrate your answer.

   (5 marks)

2. a. Describe how you implemented inheritance in one of the examples provided in the course. How does inheritance promote code reuse?

   (10 marks)

   b. How did the course demonstrate the use of interfaces in Java? Provide an example of an interface you created during the course and explain its purpose.

   (10 marks)

3. a. Explain how encapsulation is achieved in Java. Why is encapsulation important in software development?

   (5 marks)

   b. In the course, you worked with access modifiers like `private`, `protected`, and `public`. Describe a scenario where each would be appropriately used.

   (5 marks)

4. a. How did the course explain the concept of abstraction? Give an example of how you applied abstraction in one of your projects.

   (10 marks)

   b. Provide a real-world example where polymorphism would be useful in a Java application. How was polymorphism implemented in one of the course exercises?

   (10 marks)

5. a. Discuss how exception handling is implemented in Java. What are the benefits of using exception handling in OOP?

(5 marks)

b. Provide an example of a custom exception class you might create based on a scenario from the course. How would you implement this exception in your code?

(10 marks)

6. How would you approach designing a Java application for a library management system using OOP principles? Outline the classes, objects, and interactions you would consider. (Identify key components, explain interactions and relationships among classes, consider additional features, provide sample code snippets).

(10 marks)

7. Upon completing the LinkedIn Learning course on Java Object-Oriented Programming, you should have earned a course completion badge. Attach a **screenshot or digital copy of your course completion badge** that displays your name and the date of completion. In addition, briefly describe a key concept or project from the course that you found most insightful. How did this course help you enhance your understanding of Object-Oriented Programming? Provide specific examples or code snippets to illustrate your point.

(10 marks)

Additional Information: -
The report should include a title page (including your name and ID number), a table of contents (with page numbers), conclusion, and a listing of the code (in an appendix). Marks will also be awarded for the quality of writing and the presentation of the report.

(5 marks)

Marking Scheme

Marking criteria		Marks
Technical Questions		
	1. A(OOP), B (Class and Objects)	10 Marks
	2. A(Inheritance), B (Interface)	20 Marks
	3. A, B (Encapsulation)	10 Marks
	4. A(Abstraction), B (Polymorphism)	20 Marks
	5. A, B (Exception Handling)	15 Marks
	6. Design Approach	10 Marks
	7. Course completion and Learning Outcomes	10 Marks
	8. Overall Report Presentation/Formatting (Additional Information)	5 Marks
	Total	**100 Marks**