

	School of Engineering & Technology	
	Department: SOET	Session: 2025-26
	Programme: B.Tech. , BCA, BSc.	Semester: 3rd
	Course Code: ENCS201, ENCA 203, ENBC205	Number of students:
	Course Name: Java Programming	Faculty: Mr. Vishwanil S

Assignment Number: 02

Instructions:

- Assignment needs to be submitted by given date.
- Assignment must be submitted on (<https://lms.krmangalam.edu.in/>)
- Use of ChatGPT and similar tools is strictly prohibited.
- Assignment must be written on dedicated assignment copy for Java Programming subject.
- All assignments must be prepared as per the format shared in classes
- Assignments need to be submitted by each individual.
- Total marks:5
- This assignment contributes to total 10% of internal evaluation.
- Submission Requirements: Submit individually via **GitHub** and provide the link of submission by **25th September,2025**.
- Assignments will be evaluated on the basis of the following metrics.
 - Originality
 - Correctness
 - Completeness

Sr.no	Assignment Details	COs
1.	<p>Project Title Calculator Application Using Method Overloading.</p> <p>Problem Statement: Design and implement a calculator application that demonstrates the concept of method overloading in Java. The application should allow users to perform basic arithmetic operations (addition, subtraction, multiplication, division) using overloaded methods that accept different types and numbers of parameters.</p> <p>Project Objectives:</p> <ul style="list-style-type: none"> - Understand and apply the concept of method overloading in Java - Implement overloaded methods for arithmetic operations - Use Java control structures and I/O operations - Handle different data types and parameter combinations <p>Learning Outcomes:</p>	CO1

1. Develop a deeper understanding of object-oriented programming concepts in Java
2. Gain practical experience with method overloading and polymorphism
3. Implement real-world applications using overloaded methods
4. Learn to manage and share code using GitHub

Project Instructions:

Calculator Class Design:

- **Attributes:**

No specific attributes required for this assignment

- **Methods:**

- add(int a, int b)
- add(double a, double b)
- add(int a, int b, int c)
- subtract(int a, int b)
- multiply(double a, double b)
- divide(int a, int b) (Handle divide-by-zero exception)

▪ .

5. User Interface Class:

- **Attributes:**

- Scanner object for input.

Methods:

- performAddition() : This method prompts the user to enter numbers and calls the appropriate overloaded 'add' method from the Calculator class to display the result
- performSubtraction(): This method collects two integer inputs from the user and invokes the 'subtract' method to compute and show the difference.
- performMultiplication(): This method allows the user to input two double values and uses the 'multiply' method to calculate the product.
- performDivision() : This method takes two integers from the user and calls the 'divide' method, ensuring divide-by-zero errors are handled gracefully.
 - mainMenu() – Display menu and handle user choices

Implementation Steps:

- Define the Calculator class with overloaded methods
- Create the UserInterface class to interact with users
- Implement methods to handle different operations
- Use control structures for menu navigation
- Validate inputs and handle exceptions

Sample Interaction:

Welcome to the Calculator Application!

1. Add Numbers
2. Subtract Numbers
3. Multiply Numbers
4. Divide Numbers
5. Exit

Enter your choice: 1

Enter first number: 10

Enter second number: 20

Result: 30

Evaluation Highlight Rubrics (10 points):

Criterion	Points	Description
Correct Implementation of Method Overloading	1	<ul style="list-style-type: none">- Overloaded methods correctly defined and used for arithmetic operations.- Demonstrates understanding of polymorphism.
Functionality of Calculator Operations	1	<ul style="list-style-type: none">- All operations (add, subtract, multiply, divide) work as expected.- Handles different data types and parameter combinations.
Exception Handling	0.5	<ul style="list-style-type: none">- Proper handling of divide-by-zero and invalid inputs using try-catch

			blocks.	
	User Interface and Interaction Flow	1	<ul style="list-style-type: none"> - Menu-driven interface is intuitive and responsive. - Prompts and outputs are clear and user-friendly. 	
	Code Structure and Modularity	0.5	<ul style="list-style-type: none"> - Code is well-organized into classes and methods. - Follows object-oriented principles. 	
	Input Validation and Control Structures	0.5	<ul style="list-style-type: none"> - Uses control structures effectively for menu navigation and input validation. 	
	Code Documentation and Readability	0.5	<ul style="list-style-type: none"> - Includes meaningful comments and follows naming conventions. - Code is easy to read and maintain. 	

--	--	--